Stochastic Optimization and Real-Time Scheduling in Cyber-Physical Systems

by

Lei Yang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2012 by the
Graduate Supervisory Committee:

Junshan Zhang, Chair
Cihan Tepedelenlioglu
Guoliang Xue
Lei Ying

ARIZONA STATE UNIVERSITY

December 2012

ABSTRACT

A principal goal of this dissertation is to study stochastic optimization and real-time scheduling in cyber-physical systems (CPSs) ranging from real-time wireless systems to energy systems to distributed control systems. Under this common theme, this dissertation can be broadly organized into three parts based on the system environments.

The first part investigates stochastic optimization in real-time wireless systems, with the focus on the deadline-aware scheduling for real-time traffic. The optimal solution to such scheduling problems requires to explicitly taking into account the coupling in the deadline-aware transmissions and stochastic characteristics of the traffic, which involves a dynamic program that is traditionally known to be intractable or computationally expensive to implement. First, real-time scheduling with adaptive network coding over memoryless channels is studied, and a polynomial-time complexity algorithm is developed to characterize the optimal real-time scheduling. Then, real-time scheduling over Markovian channels is investigated, where channel conditions are time-varying and online channel learning is necessary, and the optimal scheduling policies in different traffic regimes are studied.

The second part focuses on the stochastic optimization and real-time scheduling involved in energy systems. First, risk-aware scheduling and dispatch for plug-in electric vehicles (EVs) are studied, aiming to jointly optimize the EV charging cost and the risk of the load mismatch between the forecasted and the actual EV loads, due to the random driving activities of EVs. Then, the integration of wind generation at high penetration levels into bulk power grids is considered. Joint optimization of economic dispatch and interrupt-

ible load management is investigated using short-term wind farm generation forecast.

The third part studies stochastic optimization in distributed control systems under different network environments. First, distributed spectrum access in cognitive radio networks is investigated by using pricing approach, where primary users (PUs) sell the temporarily unused spectrum and secondary users compete via random access for such spectrum opportunities.The optimal pricing strategy for PUs and the corresponding distributed implementation of spectrum access control are developed to maximize the PU's revenue. Then, a systematic study of the nonconvex utility-based power control problem is presented under the physical interference model in ad-hoc networks. Distributed power control schemes are devised to maximize the system utility, by leveraging the extended duality theory and simulated annealing.

To my family.

ACKNOWLEDGEMENTS

I would like to express utmost gratitude and appreciation to my advisor, Professor Junshan Zhang, for his kind support and constant encouragement. His enthusiastic attitude, unquenchable thirst and excitement towards research always drive me to work the best of mine. Moreover, he is very kind and always accessible to discuss technical and non-technical issues with me. He has been a great philosopher, friend and guide to me during my PhD study. I am very fortunate to have him as my advisor.

I would like to express my sincere gratitude to my committee members, Professors Cihan Tepedelenlioglu, Guoliang Xue, and Lei Ying. I also want to thank Professors Vijay Vittal, Tolga M. Duman and Joseph Hui. Their diverse backgrounds, inspiring suggestions and in-depth discussions have formed an intellectual environment for interdisciplinary research.

Many thanks to my current and former colleagues: Dong Zheng, Weiyan Ge, Qinghai Gao, Chandrashekhar Thejaswi, Shanshan Wang, Miao He, Dajun Qian, Xiaowen Gong, Brian Proulx, Tuan Tran, Sugumar Murugesan, Mojgan Hedayati, David Ganger, Yang Cao, Kai Cai, Ling Tang, Huilan Zou, Jia Lou, Shibo He, Chuan Huang, Donghoon Shin, Xu Chen, Zhengyu Zhang and Mengyuan Zhang for the pleasant and inspiring discussions. I also thank my collaborators: Yalin Sagduyu, Jason Li, Mung Chiang, Hongseok Kim and Chee Wei Tan for the insightful discussions I had with them.

Finally, I would like to thank my parents and my wife for their endless love and support, without whom I cannot have come this far. Especially, I am very grateful to my lovely daughter, who brings so much happiness to me and makes my PhD study easier.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview

As the next generation of engineered systems, cyber-physical systems (CPSs) have drawn a great deal of attention from academia, industry, and the government, due to its potential benefits to society, economy, and the environment. As a whole, CPSs are characterized by a tight combination of, and coordination among, computing, communication, and control technologies, to achieve stability, performance, reliability, robustness, and efficiency in physical systems [1]. The potential benefits of CPSs trigger the new design and development of many societally critical application domains such as communication, transportation, energy, and medical systems, aiming at making the systems more smart, reliable, secure, efficient, and robust. To this end, computing and networking, e.g., ubiquitous embedded computing, sensing, and wireless networking technologies, are becoming the key driving forces and components of CPSs, ranging from real-time wireless systems to energy systems to distributed control systems. Needless to say, there are many challenges due to the unique features and requirements of CPSs such as the stochastic nature of communication systems and delay requirements in real-time systems. Different from earlier generation control systems, today's CPSs are distinguished by several important characteristics: 1) the much larger scale of CPSs, 2) the heterogeneous running environments of CPSs, and 3) the more complex interactions among entities of CPSs. Due to the structural and behavioral complexities, it is more challenging to design and implement a CPS. To overcome these

issues, efforts will have to span all the constituent fields, and the spectrum of research fields relevant to CPSs is very broad. This dissertation studies stochastic optimization and real-time scheduling in CPSs, with the focus on real-time wireless systems, energy systems and distributed control systems.

1.1.1 Real-time Wireless Systems

CPSs rely on an underlying communication network to transport data packets between sensors and computational units. For real-time communication, these packets need to be delivered within a time deadline, and a certain minimum throughput of such packets is often required. Thus, CPSs need a real-time communication network that can guarantee quality of service (QoS) in terms of the throughput and delay of each flow.

Wireless networks are increasingly used to serve real-time traffic that requires strict per-packet delay bounds. Different from its wireline counterpart, real-time traffic scheduling in wireless networks is particularly challenging, since wireless transmissions are subject to shadowing, fading, and interference, and thus usually unreliable. When serving real-time traffic, the scheduler needs to explicitly take into account the stochastic and heterogeneous behavior of packet losses due to failed transmissions.

Clearly, the optimal solution to such deadline scheduling problems requires to explicitly taking into account the coupling in the deadline-aware transmissions and stochastic characteristics of the traffic and wireless channels, which involves Markov decision process (MDP). MDP-based solutions, if not designed carefully, would often lead to algorithms with prohibitively high complexity. It is therefore of great interest to develop low-complexity MDP-based scheduling algorithms for real-time wireless scheduling. Chapter

2 studies the real-time scheduling with adaptive network coding over memoryless channels, where the monotonicity-based backward induction algorithm is developed, which can characterize the optimal real-time scheduling with block size adaptation in *polynomial time*.

One important characteristic of realistic wireless channels is the channel memory [2], which is often ignored in a majority of literature by assuming that in a flat fading model, the fading coefficient evolves from one slot to another independently, i.e., without any memory. Under the fading channels with memory, the channel state information is rarely available at the scheduler. Joint channel learning and scheduling is therefore needed. Chapter 3 studies real-time scheduling over Markovian channels, for which channel conditions are time-varying and online channel learning is necessary. Downlink scheduling of multiuser traffic with hard deadlines is cast as a partially observable Markov decision process, and the optimal scheduling policies in different traffic regimes are studied.

1.1.2 Energy Systems

As one of the most important CPSs, energy systems are required to provide clean and sustainable energy generation, transmission, and distribution for the global public interest. Smart grids, the next generation infrastructure for electric power systems, are designed to produce, distribute, and use electricity in a more clean, efficient, and cost-effective manner through the integration of computing, communication, and control technologies. To this end, many challenges need to be tackled, e.g., in electrification of transportation and renewable energy integration.

Smart Charging for Electric Vehicles

Today's transportation sector accounts for a significant portion of petroleum consumption and carbon emissions worldwide. The growing concerns over energy security, the dependence on oil/petroleum and environmental issues are driving the electrification of transportation and the development of plug-in electric vehicle (EV) technology. By using electricity rather than petroleum, electric vehicles and plug-in hybrid electric vehicles reduce the petroleum consumption and the greenhouse gas emissions. The large-scale implementation of EVs is being planned in the near future. It has been widely recognized that the high penetration level of EVs would have considerable impact on the existing power system. Without proper coordination, the coincidence between peaks of EV and non-EV load will require investment in generation, transmission, and distribution, in order to maintain the reliability of the power system. However, the stochastic nature of EVs' arrival and departure makes the charging problem challenge. Chapter 4 studies the EV charging problem that minimizes the EV charging cost, while considering the random driving activities of EVs.

Renewable Energy Integration

In order to meet the renewable portfolio standards (RPS) adopted by many states in the U.S., much effort is being made to integrate renewable generation (particularly wind generation) into bulk power grids. Indeed, wind energy constitutes a significant portion of this renewable integration [3]. High penetration of wind generation, however, is expected to result in significant operational challenges [4], due to its non-dispatchability and variability. Accurate

forecasts of future wind generation across temporal and spatial scales remains still elusive [5]. Therefore, the integration of wind generation at high penetration levels into bulk power grids may have significant impact on system reliability, because of the inability to obtain an acceptable load/generation balance. Chapter 5 investigates joint optimization of economic dispatch and interruptible load management with increased wind penetration by using short-term wind farm generation forecast.

1.1.3 Distributed Control Systems

One objective of CPSs is to develop distributed control technologies that support a wide spectrum of system operation to adapt to resource constraints, such as limitations of computational, energy, and networking resources.

Distributed Spectrum Access in Cognitive Radio Networks

In communication systems, cognitive radio (CR) could benefit enormously from distributed consensus about available bandwidth and from distributed control technologies. CR is expected to capture temporal and spatial "spectrum holes" in the spectrum white space, and to enable spectrum sharing for secondary users (SUs). One grand challenge is how SUs can discover spectrum holes and access them efficiently, without causing interference to the primary users (PUs), especially when the demand for available spectrum nearly outstrips the supply. Due to the decentralized nature of cognitive radio networks, coordination among SUs remains a challenge when SUs of different networks coexist [6], simply because contention between SUs is unavoidable. Distributed spectrum access strategies are therefore required so that SUs can adapt quickly to the changing spectrum environment by dynamically adjusting operating frequency, bandwidth, and other physical layer system parameters.

To this end, random access can serve as a platform for the contention among SUs, and can be employed for distributed spectrum access. Chapter 6 takes the pricing approach to study the spectrum access control in cognitive radio networks, and develops decentralized spectrum access control mechanisms for two spectrum trading market models based on random access: one with a monopoly PU market and the other with a multiple PU market.

Power Control in Wireless Networks

Power control has been extensively studied as an important mechanism to control Signal-to-Interference-plus-Noise Ratios (SINR), which in turn determines QoS metrics such as rate, outage, and delay. One primary objective of power control is to maximize the system utility that can achieve a variety of fairness objectives among users [7–10]. However, maximizing the system utility, under the physical interference model, often involves nonconvex optimization and it is known to be NP-hard, due to the complicated coupling among users through interference [11]. Besides, since ad-hoc networks have no central controller such as the base station, distributed algorithms are often required for power control. Therefore, it is challenging to find the globally optimal power allocation in a distributed manner.

Most existing works cannot find the globally optimal power allocation in a distributed setting. Their solutions [12–16] are either distributed but suboptimal or optimal but centralized. Recently, an interesting work [17] has proposed the SEER algorithm based on Gibbs sampling [18], which can approach the globally optimal solution in an asymptotic sense when the control parameter in Gibbs sampling tends to infinity. Notably, for each iteration in the SEER algorithm, each user utilizes Gibbs sampling to compute its tran-

sition probability distribution for updating its transmission power, where the requirement for message passing and computing the transition probability distribution in each iteration can be demanding when applied to ad-hoc networks. A challenging task in distributed power control in ad-hoc networks is to reduce the amount of message passing while preserving the global optimality. Chapter 7 tackles this challenge by combining recent advances in extended duality theory (EDT) [19] with simulated annealing (SA) [20], and develops the distributed stochastic power control algorithm that can achieve the global optimality in a distributed manner.

1.2 Summary of Main Contributions

The main body of this dissertation can be categorized into three parts. The first part (Chapters 2-3) studies the scheduling problems in the real-time wireless systems by using dynamic programming. In particular, the optimal scheduling over memoryless and Markovian channels is investigated respectively. The second part (Chapters 4-5) studies the EV charging and the wind integration in the energy systems. Chapter 4 studies the EV charging problem that minimizes the EV charging cost, while considering the random driving activities of EVs. Chapter 5 investigates joint optimization of economic dispatch and interruptible load management with increased wind penetration by using short-term wind farm generation forecast. The third part (Chapters 6-7) studies the distributed optimization of network resource allocation in distributed control systems. Specifically, Chapter 6 takes the pricing approach to study the spectrum access control in cognitive radio networks, and develops decentralized spectrum access control mechanisms for two spectrum trading market models. Chapter 7 studies the nonconvex case of utility-based pow-

er allocation in wireless ad-hoc networks, and develops distributed stochastic power control algorithms by using the extended duality theory and simulated annealing. A brief summary of the main contributions in each chapter in the rest of the dissertation is given as follows.

Chapter 2 studies the real-time scheduling with adaptive network coding (NC) over memoryless channels. To meet hard deadlines of real-time traffic, the block size for NC is adapted based on the remaining time to the deadline so as to strike a balance between maximizing the throughput and minimizing the risk that the entire block of coded packets may not be decodable by the deadline. This sequential block size adaptation problem is then cast as a finite-horizon Markov decision process. One interesting finding is that the optimal block size and its corresponding action space monotonically decrease as the deadline approaches, and that the optimal block size is bounded by the "greedy" block size. These unique structures make it possible to significantly narrow down the search space of dynamic programming, building on which a monotonicity-based backward induction algorithm (MBIA) is developed, which can find the *optimal* block size in *polynomial time*. Further, a joint real-time scheduling and channel learning scheme with adaptive NC is developed to adapt to channel dynamics in a mobile network environment. Then, the analysis is generalized to multiple flows with hard deadlines and long-term delivery ratio constraints. A low-complexity online scheduling algorithm integrated with the MBIA is devised, and then its asymptotical utility optimality is established. The analysis and simulation results are corroborated by high fidelity wireless emulation tests, where actual radio transmissions over emulated channels are performed to demonstrate the feasibility of the MBIA in finding the optimal block size in real time.

Chapter 3 studies the real-time scheduling over Markovian channels, for which channel conditions are time-varying and online channel learning is necessary. Downlink scheduling of multiuser traffic with hard deadlines and packet-level priorities is cast as a partially observable Markov decision process. User channels are modeled as Markovian and the base station can learn only the channel condition of the currently scheduled user. The optimization of joint channel learning and scheduling presents the combined challenges incurred by the strict deadline constraint of real-time traffic and the partial observability of multiuser channels. In particular, it is shown that *idling* adds a new dimension to the action space; and that, through a case study of heterogeneous multiuser networks, idling is indeed the optimal action under certain system states in the light traffic regime. This somewhat surprising result reveals the existence of the fundamental tradeoffs between exploitation and exploration/idling, going beyond the classic "exploitation vs exploration". Due to hard deadlines and packet priorities, idling is found intimately related to the tradeoff between the successful transmission of backlogged packets and that of future arrivals. In contrast, for the special case with a symmetric two-user system, it is shown that the scheduling problem exhibits unique structures, rendering a non-idling greedy policy optimal.

Chapter 4 studies risk-aware day-ahead scheduling and real-time dispatch for plug-in electric vehicles (EVs), aiming to jointly optimize the EV charging cost and the risk of the load mismatch between the forecasted and the actual EV loads, due to the random driving activities of EVs. It turns out that the inclusion of the load mismatch risk in the objective function complicates the risk-aware day-ahead scheduling and indeed the optimization problem is nonconvex. A key step is to utilize the *hidden convexity* structure

to recast it as a two-stage stochastic linear program, which can be solved by using the L-shaped method. Further, a distributed risk-aware real-time dispatch algorithm is developed, where the aggregator only needs to compute the shadow prices for each EV to optimize its own charging strategy in a distributed manner. It is shown, based on real data, that the proposed risk-aware day-ahead scheduling algorithm can reduce not only the overall charging cost, but also the peak demand of EV charging.

Chapter 5 studies joint optimization of economic dispatch (ED) and interruptible load management by using short-term wind farm generation forecast. Specifically, a finite state Markov chain model for wind farm generation forecast is developed based on spatial and temporal charactersitics of wind turbine power outputs. In light of the Markovian property of the proposed forecast model, the joint optimization of ED and interruptible load management is cast as a Markov decision process (MDP) based dynamic programming problem. To reduce the complexity of this joint optimization problem, a "greedy" policy is used, where "greedy" refers to minimizing the immediate system cost only. Further, by leveraging the convexity properties of the formulated problem, the proposed ED problem is transformed into a stochastic program by using the point forecast of wind farm generation. Numerical studies, via using the IEEE Reliability Test System – 1996 and realistic wind measurement data from an actual wind farm, demonstrate the significant benefits obtained by integrating the Markov-chain-based forecast and the interruptible load management, compared with conventional wind-speed-based forecasting methods.

Chapter 6 studies pricing-based spectrum access control in cognitive radio networks, where primary users (PUs) sell the temporarily unused spectrum and secondary users (SUs) compete via *random access* for such spectrum

opportunities. Compared with existing market-based approaches with centralized scheduling, pricing-based spectrum management with random access provides a platform for SUs contending for spectrum access and is amenable to distributed implementation due to its low complexity. Two market models, one with a monopoly PU market and the other with a multiple PU market, are considered. For the monopoly PU market model, distributed pricing-based spectrum access mechanisms are devised, which enables SUs to contend for channel usage. Specifically, SUs contending via slotted Aloha is considered first. Since the revenue maximization problem therein is nonconvex, the corresponding Pareto optimal region is first characterized, and a Pareto optimal solution is then obtained, which maximizes the SUs' throughput subject to their budget constraints. To mitigate the spectrum under-utilization due to the "price of contention," the problem is revisited, where SUs contend via C-SMA, which results in more efficient spectrum utilization and higher revenue. Then, the tradeoff between the PU's utility and its revenue is studied, when the PU's salable spectrum is controllable. Next, for the multiple PU market model, the competition among PUs is cast as a three-stage Stackelberg game, where each SU selects a PU's channel to maximize its throughput. The existence and the uniqueness of Nash equilibrium, in terms of access prices and the spectrum offered to SUs, are explored and an iterative algorithm is developed for strategy adaptation to achieve the Nash equilibrium. Our findings reveal that there exists a unique Nash equilibrium when the number of PUs is less than a threshold determined by the budgets and elasticity of SUs.

Chapter 7 studies the nonconvex case of utility-based power allocation in wireless ad-hoc networks. To tackle this challenge, it is first shown that the globally optimal point lies on the boundary of the feasible region, which

is utilized as a basis to transform the utility maximization problem into an equivalent max-min problem with more structure. By using extended duality theory, penalty multipliers are introduced for penalizing the constraint violations, and the minimum weighted utility maximization problem is then decomposed into subproblems for individual users to devise a distributed stochastic power control algorithm, where each user stochastically adjusts its target utility to improve the total utility by simulated annealing. The proposed distributed power control algorithm can guarantee global optimality at the cost of slow convergence due to simulated annealing involved in the global optimization. The geometric cooling scheme and suitable penalty parameters are used to improve the convergence rate. Next, by integrating the stochastic power control approach with the back-pressure algorithm, a joint scheduling and power allocation policy is developed to stabilize the queueing systems. Finally, the above distributed power control algorithms are generalized to multicast communications, and their global optimality for multicast traffic is shown.

Chapter 8 summarizes the dissertation along with a discussion on future directions of the research.

Chapter 2

JOINT SCHEDULING AND ADAPTIVE NETWORK CODING FOR REAL-TIME TRAFFIC OVER MEMORYLESS CHANNELS

2.1 Introduction

The past few years have witnessed a tremendous growth of multimedia applications in wireless systems. To support the rapidly growing demand in multimedia traffic, wireless systems must meet the stringent quality of service (QoS) requirements, including the minimum bandwidth and maximum delay constraints. However, the time-varying nature of wireless channels and the hard delay constraints give rise to great challenges in scheduling multimedia traffic flows. In this chapter, we study the efficient use of *network coding* (NC) to optimize the throughput of multimedia traffic over wireless channels under the hard deadline constraint.

In capacitated multihop networks, NC is known to optimize the multicast flows from a single source to the min-cut capacity [21]. NC also provides coding diversity over unreliable wireless channels and improves the throughput and delay performance of single-hop broadcast systems, compared to (re)transmissions of uncoded packets [22–28]. Nevertheless, the block NC induces "decoding delay," i.e., receivers may not decode network-coded packets until a sufficient number of innovative packets are received. Therefore, the minimization of NC delay has received much attention (e.g., [29–33]).

For multimedia traffic, meeting the deadline may be more critical than reducing the average delay. Under the *hard deadline* constraints, NC may result in significant performance loss, unless the receivers can decode the packets

before the deadline. Different NC mechanisms (e.g., [34–37]) have been proposed recently to incorporate deadline constraints. An immediately-decodable network coding (IDNC) scheme has been proposed in [34] to maximize the broadcast throughput subject to deadlines. A partially observable Markov decision process (POMDP) framework has been proposed in [35] to optimize media transmissions with erroneous receiver feedback.

These works focus on optimizing network codes in each transmission; however, such an approach is typically not tractable due to the "curse of dimensionality" of dynamic programming. To reduce the complexity of optimizing network codes in each transmission, [36] has formulated a joint coding window selection and resource allocation problem to optimize the throughput in deadline-constrained flows. However, the computational complexity can be still overwhelming due to the finite-horizon dynamic programming involved in the coding window selection. To overcome this limitation, [36] has proposed a heuristic scheme with fixed coding window to tradeoff between the optimality and complexity objectives.

A primary objective of this study is to 1) develop optimal adaptive NC schemes with low computational complexity, extended to joint scheduling and NC block size adaptation for multiple flows, and 2) integrate channel learning with adaptive NC over wireless broadcast erasure channels and demonstrate the feasibility of the analytical solutions with hardware-in-the-loop wireless emulation tests. Our main contributions are summarized as follows.

- An adaptive NC scheme that sequentially adjusts the block size (coding block length) of NC is developed to maximize the system throughput, subject to the hard deadlines (cf. [36]). This sequential block size

adaptation problem is formulated as a finite-horizon Markov decision process, where the underlying dynamic programming solutions would typically result in exponential size of search space and may not render any practical solution in general. We show that the optimal block size and its corresponding action space monotonically decrease as the packet deadline approaches, and the optimal block size is bounded by the "greedy" block size that maximizes the immediate throughput only. These unique structures make it possible to narrow down the search space of dynamic programming, and accordingly a monotonicity-based backward induction algorithm (MBIA) is developed to solve for the optimal block size in *polynomial time*, compared with [35, 36]. Depending on the remaining time to deadline, the block size is adapted over time to maximize the expected throughput (including immediate and future returns in case more time is left for transmissions). Additionally, a joint real-time scheduling and *channel learning* scheme with adaptive NC is developed for the practical case, in which the scheduler does not have (perfect) channel information such as the case of a mobile network with time-varying channels.

- The study on adaptive NC is generalized to the case with multiple flows. A joint scheduling and block size adaptation approach is developed to maximize the system utility subject to the long-term delivery ratio and the hard-deadline constraint of each flow. By integrating the MBIA in the model with multiple flows, a low-complexity online scheduling algorithm is constructed to dynamically control the flow rate and system resource allocated to each flow. This online algorithm is shown to achieve a utility that is within $O(\rho)$ of the optimal solution for any step size

$\rho \geq 0$, while satisfying the long-term delivery ratio and the hard-deadline constraint of each flow.

- The adaptive NC schemes are implemented in a realistic wireless emulation environment with real radio transmissions. Our high fidelity testbed results corroborate the feasibility of the MBIA in finding the optimal block size in real time. As expected, the adaptive NC scheme with the MBIA outperforms the fixed coding scheme, and the proposed scheme of joint real-time scheduling and channel learning performs well under unknown and dynamic channel conditions.

The rest of the chapter is organized as follows. In Section 2.2, the system model and the block size adaptation problem with the hard deadlines are presented respectively. In Section 2.3, the MBIA is developed to solve for the optimal block size and building on this, the joint real-time scheduling and channel learning scheme with adaptive NC is devised for the case with unknown channel information. In Section 2.4, the study on adaptive NC is generalized to multiple flows. In Section 2.5, the adaptive NC schemes are implemented and tested in a realistic wireless emulation environment with hardware-in-the-loop experiments. The chapter is concluded in Section 2.6.

2.2 Throughput Maximization vs. Hard Deadline

2.2.1 System Model

We consider a time-slotted downlink system with one transmitter (e.g., base station) and N receivers (users), as illustrated in Fig. 2.1. Time slots are synchronized across receivers and the transmission time of a packet corresponds to one time slot. The transmitter broadcasts M packets to N receivers over

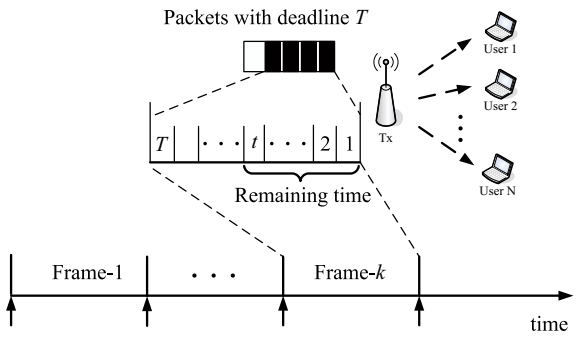

Figure 2.1: System model. (The arrow denotes the time instant for drops of undelivered packets and arrivals of new packets.)

i.i.d. binary erasure channels with erasure probability ϵ.[1] For multimedia communications, it is standard to impose deadlines for delay-sensitive data (see, e.g., [27, 34–36, 38]). We assume that packets must be delivered to each receiver before T slots, i.e., the deadline of each packet is T slots. Any packet that cannot be delivered to all receivers by this deadline is dropped without contributing to the throughput.

Worth noting is that this model can be readily applied to finite-energy systems with NC, where the objective is to maximize the system throughput before the energy is depleted for further transmission. Therefore, the energy and delay constraints can be used interchangeably.

In Section 2.3, the basic model with one flow and one frame of T slots is considered. In Section 2.4, the model is generalized to multiple frames with multiple flows, where packets arrive at the beginning of each frame and they are dropped if they cannot be delivered to their receivers by the deadline of T slots.

[1]The results derived in the chapter can be readily applied to heterogeneous channels with different erasure probabilities.

2.2.2 Network Coding for Real-time Scheduling

As noted above, the throughput gain of NC comes at the expense of longer decoding delay (since packets are coded and decoded as a block), which may reduce the throughput of the system due to the hard deadline constraints. At the beginning of each frame, a block of network-coded packets is transmitted. If this coded block is decoded before the end of a frame, a new block is initiated until the end of the frame. As we will show in Section 2.3.4, this sequential NC block size adaptation policy provides significant gains over the intuitive but suboptimal approach that would select the block size in such a way that it takes up most of the frame on average, leaving little opportunity for additional transmissions.

Let K denote the block size, i.e., the number of original packets encoded together by NC. We assume that the transmitter and each receiver know the set of coding coefficients, and the transmitter broadcasts the value of K to receivers before the NC transmissions start. Linearly independent coding coefficients are used. The coding coefficients can also be chosen randomly from a large field size (or from a predetermined coding coefficient matrix of rank K) such that with high probability K packet transmissions deliver K innovative packets in coded form to any receiver, i.e., the entire block of packets can be decoded after K successful transmissions. Immediate and perfect feedback is assumed to be available at the transmitter such that each receiver sends acknowledgement to transmitter not per packet or slot basis but once after the block of K packets is successfully decoded.

As shown in [23], the probability that all receivers can decode the block

of size K within T slots is given by

$$P(K,T) = \left(\sum_{\tau=K}^{T} \binom{\tau-1}{K-1} \epsilon^{\tau-K} (1-\epsilon)^{K} \right)^{N}, \tag{2.1}$$

where $\binom{n}{m}$ denotes the number of combinations of size m out of n elements.[2] Note that (2.1) strongly depends on the choice of block size K and we show that,

Lemma 2.2.1. *The decoding probability (2.1) decreases monotonically with K for fixed T.*

Proof. The proof is given in Appendix 2.7.1. □

With block NC, there is the risk that none of the packets can be decoded by the receivers before the hard deadline. By using IDNC, it may be possible to start decoding without waiting for the entire block to arrive but the complexity of finding a suitable code may be overwhelming due to the dynamic programming involved in the problem [34]. Here, we provide the throughput guarantees for the worst-case scenario, where either the whole block or none of the packets can be decoded at any slot. There is a tradeoff between the block size and the risk of decoding. In particular, we cannot greedily increase K to maximize the system throughput under the hard deadline constraints, since the risk that some receivers cannot decode the packets, i.e., $1-P(K,T)$, also increases with K according to Lemma 2.2.1.

If the first block is delivered within the deadline, i.e., T slots, the size of a new block (with new packets) needs to be re-adjusted for the remaining slots. In other words, we need *real-time* scheduling of network-coded transmissions

[2]We can also employ random NC with a finite field size q. This would change the decoding probability (2.1) to a function of q. However, the general structure of the results will remain the same.

depending on how close the deadline is. For example, when there is only one slot left before the deadline, the optimal block size is 1, since for any $K > 1$, no receivers can decode the packets before the deadline. Also, the block size in a given slot statistically determines the remaining slots (before the deadline) along with the future system throughput. In Section 2.3, the optimal block size adaptation policy is derived to maximize the system throughput under the deadline constraints for one frame with one flow. In Section 2.4, the results are generalized to the case with multiple frames and multiple flows.

2.2.3 Problem Formulation: A Markov Decision Process View

The NC-based multimedia traffic scheduling of one frame is a sequential decision problem, which can be cast as a Markov decision process (MDP) described as follows.

Horizon: The number of slots available before the deadline over which the transmitter (scheduler) decides the block size is the horizon. Due to the hard deadline, this MDP problem is a finite horizon problem with T slots (one frame).

State: The remaining slots $t \in \{0, 1, ..., T\}$ before the hard deadline is defined as the state,[3] where $t = 0$ denotes that there is no slot left for transmissions.

Action: Let K_t, $t \in \{1, ..., T\}$, denote the action taken at state t, which is the block size for the remaining t slots. Let M_t denote the number of packets not delivered yet at state t. Thus, at state $t > 0$, K_t can be chosen from 1 to $\min(t, M_t)$. For $t = 0$, the transmitter stops transmitting any packet, i.e., $K_0 = 0$. In general, the action space is defined as $\mathcal{K}_t = \{0, 1, ..., \min(t, M_t)\}$.

[3]We use the terms "state" and "slot" interchangeably.

Expected immediate reward: For the remaining t slots, the expected immediate reward is the expected number of packets successfully decoded by all receivers, which is given by

$$R_t(K_t) = K_t\, P(K_t, t), \tag{2.2}$$

where $P(K_t, t)$ denotes the probability that each receiver can decode these K_t packets within t slots and it is given by (2.1).

Block size adaptation policy: A block size adaptation policy $\mathcal{P}$ is a sequence of mappings, $\mathcal{P} = \{\mathcal{P}_t\}_{t=1}^{T}$, from t, M_t, ϵ, and N to an action $K_t \in \{0, 1, ..., \min(t, M_t)\}$, i.e., $K_t = \mathcal{P}_t(t, M_t, \epsilon, N) = \min(\mathcal{P}_t(t, \epsilon, N), M_t)$. Without loss of generality, in Section 2.3, we assume that M_t is always larger than t, i.e., $K_t \in \{0, 1, ..., t\}$. This does not change the monotonic structure of the block size with state t. We will discuss these structural properties in detail in Section 2.3.

Total expected reward: Given the adaptation policy $\mathcal{P}$, the total expected reward for the remaining t slots is given by

$$\begin{aligned} V_t(K_t; \mathcal{P}) &= R_t(K_t) + E[V_j(K_j; \mathcal{P})] \\ &= R_t(K_t) + \sum_{j=0}^{t-K_t} q_t(j) V_j(K_j; \mathcal{P}), \end{aligned} \tag{2.3}$$

where the probability mass function $q_t(j) = P(K_t, t-j) - P(K_t, t-j-1)$ denotes the probability that the block of size K_t is delivered over exactly j slots before the deadline.

2.3 Network Coding with Adaptive Block Size

One main thrust of this chapter is devoted to the development and analysis of the polynomial-time monotonicity-based backward induction algorithm (M-

BIA). The design of the MBIA is motivated by the structures of the optimal and the greedy policies that are formally defined as follows.

Definition 2.3.1. *A real-time scheduling policy with adaptive network coding is optimal, if and only if it achieves the maximum value of the total expected reward given by the Bellman equation [39] in dynamic programming:*

$$V_t(K_t^*;\mathcal{P}^*) = \max_{K_t\in\{0,1,\ldots,t\}} \{R_t(K_t) + \sum_{j=0}^{t-K_t} q_t(j)V_j(K_j^*;\mathcal{P}^*)\}, \tag{2.4}$$

where K_t^* denotes the optimal block size, $\mathcal{P}^*$ denotes the optimal block size adaptation policy, and the terminal reward is given by $V_0(0;\mathcal{P}^*) = 0$.

Definition 2.3.2. *The greedy policy maximizes only the expected immediate reward (2.2) without considering the future rewards and the greedy decision is given by*

$$\hat{K}_t = \arg\max_{K_t\in\{0,1,\ldots,t\}} R_t(K_t). \tag{2.5}$$

2.3.1 Optimal Block Size Adaptation Policy

In each slot t, the optimal policy balances the immediate reward and the future reward by selecting a suitable block size K_t^*. In general, the approach of solving for the optimal block size by traditional dynamic programming [39] suffers from the "curse of dimensionality," where the complexity of computing the optimal strategy grows exponentially with t. However, the optimal block size and its corresponding action space exhibit several monotonicity structures, and the optimal block size is bounded by the greedy block size. These unique structures make it possible to narrow down the search space of dynamic programming, and accordingly we develop a monotonicity-based backward induction algorithm (MBIA) with polynomial time complexity.

The MBIA searches for the optimal block size by backward induction and provides the optimal block size for each system state. Depending on

the remaining time to deadline, the scheduler transmits coded packets with the optimal block size until each user receives enough packets to decode this block. Then, the scheduler adjusts the block size based on the current state, and proceeds with the new block transmission. This continues until the packet deadline expires or all packets are delivered. We present next the structural properties of block size adaptation problem that will lead to the formal definition of the MBIA.

Lemma 2.3.1. *The action space $\mathcal{K}_t$ monotonically shrinks as t decreases.*

As the number of remaining slots t decreases, the maximum possible block size decreases as well, since $K_t \in \{0, 1, ..., t\}$; otherwise no receiver can decode the block of coded packets.

Proposition 2.3.1. *The expected immediate reward function $R_t(K_t)$ has the following properties:*

1. *$R_t(K_t)$ is unimodal for $K_t \in \{0, 1, ..., t\}$.*[4]

2. *$\hat{K}_t$ in (2.5) monotonically decreases as t decreases.*

Proof. The proof is given in Appendix 2.7.2. □

Fig. 2.2 shows the possible curves of $R_t(K_t)$ for different values of t, illustrating the unimodal property of $R_t(K_t)$ formally stated in Proposition 2.3.1. Based on Proposition 2.3.1, the monotonicity property of the optimal block size K_t^* is given by the following theorem.

[4] $f(x)$ is a unimodal function if for some m, $f(x)$ is monotonically increasing for $x \leq m$ and monotonically decreasing for $x \geq m$. The maximum value is attained at $x = m$ and there are no other local maximum points.

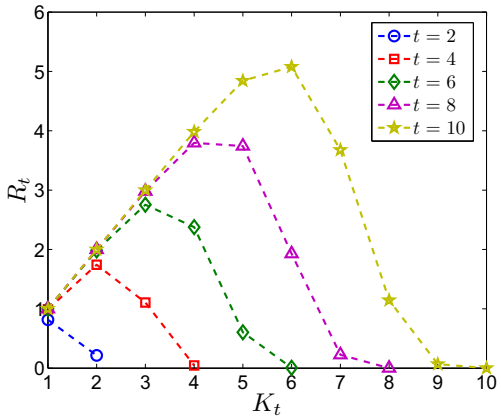

Figure 2.2: The unimodal property of $R_t(K_t)$.

Theorem 2.3.1. *The optimal block size K_t^* decreases monotonically with t, i.e., $K_t^* \geq K_{t-1}^*$, for any t.*

Proof. The proof is given in Appendix 2.7.3. □

As t decreases, the risk that some receivers cannot decode the given block of packets increases for a fixed block size. Therefore, the scheduler becomes more conservative in the block size adaptation and selects a smaller block size for smaller t.

Remarks: 1) For $t = 1, 2$, the optimal block size is $K_t^* = 1$, which can be obtained by computing the Bellman equation (2.4). 2) When $N = 1$, the optimal block size is $K_t^* = 1$ for all t, since the plain retransmission policy with $K_t = 1$ is better than the block coding with $K_t > 1$ in the presence of the hard deadlines.

Theorem 2.3.2. *The optimal block size K_t^* is no greater than the greedy block size $\hat{K}_t$ for any t.*

Proof. The proof is given in Appendix 2.7.4. □

Corollary 2.3.1. *At state t, if $R_t(K_t) > R_t(K_t+1)$, then $K_j^* \leq \hat{K}_j \leq K_t$ for any $j \in \{1, ..., t\}$.*

Corollary 2.3.1 follows directly from Proposition 2.3.1 and Theorem 2.3.2. Based on these structural properties, we develop the MBIA, which is presented in Algorithm 1.

Algorithm 1 Monotonicity-based Backward Induction Algorithm (MBIA)

1) Set $t = 0$ and $V_0(0; \mathcal{P}^*) = 0$.
2) Substitute $t + 1$ for t, and compute $V_t(K_t^*; \mathcal{P}^*)$ by searching $K_t \in \mathcal{K}_t$, where $\mathcal{K}_t = \{K_{t-1}^*, K_{t-1}^* + 1, ..., \hat{K}_t\}$, i.e.,

$$V_t(K_t^*; \mathcal{P}^*) = \max_{K_t \in \mathcal{K}_t} \{R_t(K_t) + \sum_{j=0}^{t-K_t} q(j) V_j(K_j^*; \mathcal{P}^*)\},$$

and $K_t^* = \arg\max_{K_t \in \mathcal{K}_t} \{R_t(K_t) + \sum_{j=0}^{t-K_t} q(j) V_j(K_j^*; \mathcal{P}^*)\}$.
3) If $t = T$, stop; otherwise go to step 2.

The MBIA confines the search space at state t to the interval from K_{t-1}^* (the optimal policy at state $t-1$) to $\hat{K}_t$ (the greedy policy at state t). Thus, the MBIA reduces the search space over time and reduces the complexity of dynamic programming as given by the following theorem.

Theorem 2.3.3. *The MBIA is a polynomial-time algorithm and the complexity is upper bounded by $O(T^2)$.*

Proof. The proof is given in Appendix 2.7.5. □

Remarks: By using the MBIA, the optimal block size can be computed in polynomial time, which is a desirable property for online implementation. The optimal block size depends on the number of receivers and channel erasure probabilities. For different flows, the set of receivers may be different, which

may result in different optimal block sizes, even when the number of remaining slots is the same across these flows. Therefore, without using the MBIA, offline schemes would need to compute the optimal policies for all possible receiver sets; however, this would be a computationally demanding task, as the number of receivers increases.

Based on the monotonicity properties of the greedy and optimal block sizes, the optimal policy becomes the plain retransmission, if the channel erasure probability is sufficiently large. This sufficiency condition for $K_t^* = 1$ at slot t is formally given as follows.

Theorem 2.3.4. *At slot t, the optimal policy switches to the plain retransmission policy, i.e., $K_t^* = 1$, when the erasure probability satisfies the threshold condition:*

$$\epsilon > \epsilon^*(t, N), \tag{2.6}$$

where $\epsilon^*(t, N) \in (0, 1)$ is the non-trivial (unique) solution to $R_t(1) = R_t(2)$.

Proof. The proof is given in Appendix 2.7.6. □

Note that (2.6) is a sufficient condition only and indicates the optimality of the greedy policy when ϵ is large enough. Fig. 2.3 depicts how the threshold ϵ^* varies with t and N. The underlying monotonicity property is formally stated in Corollary 2.3.2.

Corollary 2.3.2. *The threshold $\epsilon^*(t, N)$ increases monotonically with t and decreases monotonically with N.*

Remarks: 1) When the channel is good enough (with $\epsilon < \epsilon^*$), NC with $K_t > 1$ can always improve the throughput compared to the plain retransmission policy. 2) As t increases (i.e., the deadline becomes looser), the

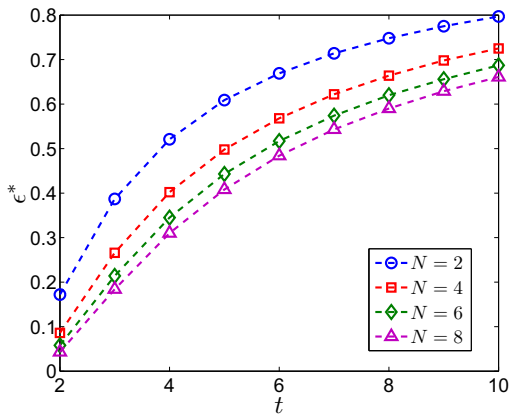

Figure 2.3: The monotonicity property of ϵ^*.

risk of decoding network-coded packets decreases, i.e., $\epsilon^*(t, N)$ increases. 3) As N increases, it becomes more difficult to meet the deadline for each of N receivers and therefore $\epsilon^*(t, N)$ drops accordingly.

2.3.2 Robustness vs. Throughput

The focus of real-time scheduling problem presented so far has been on the expected throughput without considering the variation from the average performance. Therefore, it is possible that the instantaneous throughput drops far below the expected value. To reduce this risk, we use additional variation constraints to guarantee that the throughput performance remains close to the average. In particular, for each slot t, we introduce the variation constraint to the block size adaptation problem as follows:

$$v_t(K_t) < \sigma_t^2, \ \forall K_t \in \mathcal{K}_t, \tag{2.7}$$

where σ_t^2 is the maximum variation allowed in slot t and the performance variation $v_t(K_t)$ under action K_t is given by

$$v_t(K_t) = \sum_{i=1}^{\infty} i^2 (P(K_t, i) - P(K_t, i-1)). \tag{2.8}$$

Since $v_t(K_t)$ increases with K_t, (2.7) can be rewritten as the maximum block size constraint for each slot t, i.e.,

$$K_t \leq K_t^{\max}, \tag{2.9}$$

where $K_t^{\max} = \max\{K_t | K_t = \lfloor v_t^{-1}(\sigma_t) \rfloor\}$, $v_t^{-1}(\cdot)$ is the inverse mapping of $v_t(\cdot)$, and $\lfloor x \rfloor$ denotes the largest integer smaller than x. The variation constraints do not change the monotonicity property of the optimal block size provided by Theorem 2.3.1. By introducing the variation constraints (2.9), the scheduler becomes more conservative in the block size adaptation. The additional bound $K_t^{\max}$ can be easily incorporated into the MBIA by changing the action space to $\mathcal{K}_t = \{K_{t-1}^*, K_{t-1}^* + 1, ..., \min(K_t^{\max}, \hat{K}_t)\}$ at state t.

2.3.3 Block Size Adaptation under Unknown Channels

So far we have discussed the real-time scheduling policies with adaptive NC, where the channel erasure probability ϵ is perfectly known to the scheduler. The throughput performance of these policies depends on ϵ; therefore, the scheduler needs to learn ϵ while adapting the block size, when it does not have (perfect) channel knowledge. Let $\hat{\epsilon}_t$ denote the estimate of the channel erasure probability in slot t. The scheduler can update $\hat{\epsilon}_t$ based on the feedback from the receivers. In slot T, if $\hat{\epsilon}_T < \epsilon$, we would expect with high probability that a block of packets with the size that is calculated with respect to $\hat{\epsilon}_T$ cannot be delivered before the deadline. Therefore, it is better to select the block size conservatively at the beginning, when the estimate $\hat{\epsilon}_t$ cannot be highly accurate yet, because of the small number of samples. As $\hat{\epsilon}_t$ improves over time, the block size can be gradually increased to improve the system throughput. Once the estimate is close enough to the actual value of ϵ after enough samples are collected, the block size should be adjusted (and reduced

over time) according to the MBIA.

Clearly, there is a tradeoff between the channel learning and the block size adaptation. Here, we formulate a joint real-time scheduling and channel learning algorithm (Algorithm 2) to adapt the block size while updating the maximum likelihood estimate $\hat{\epsilon}_t$ of channel erasure probability. In slot t, based on the feedback, the scheduler can compute the packet loss ratio, $\epsilon_t = 1 - \frac{n_t}{N}$, where n_t denotes the number of receivers that successfully receive a packet in slot t. Accordingly, the estimated channel erasure probability $\hat{\epsilon}_t$ is given by the moving average

$$\hat{\epsilon}_t = \frac{(T-t)\hat{\epsilon}_{t+1}+\epsilon_t}{T-t+1}. \tag{2.10}$$

The scheduler decides on the block size by comparing the temporal variation $|\hat{\epsilon}_t - \hat{\epsilon}_{t+1}|$ with a threshold δ. A detailed description is given in Algorithm 2.

Algorithm 2 Joint Real-time Scheduling and Channel Learning with Adaptive Network Coding

Initialization: Choose threshold δ and set $K_T = 1$.
Repeat until $t = 0$.
  Update channel estimate $\hat{\epsilon}_t$ by (2.10).
  Compute block size K_t^* by Algorithm 1 with $\hat{\epsilon}_t$.
  If $|\hat{\epsilon}_t - \hat{\epsilon}_{t+1}| > \delta$ **then**
    If $K_t^* \geq K_{t+1} + 1$ **then**
      $K_t = K_{t+1} + 1$,
    Else
      $K_t = K_{t+1}$.
    Endif
  Else
    $K_t = K_t^*$.
  Endif

Remarks: Algorithm 2 captures the tradeoff between the channel learning and block size adaptation. There are two options for the scheduler depending on the relationship between $|\hat{\epsilon}_t - \hat{\epsilon}_{t+1}|$ and δ. If the channel estimation is not yet good enough, Algorithm 2 chooses the block size conservatively

by incrementing K_t by at most 1. Otherwise, Algorithm 2 computes the block size by applying the MBIA.

2.3.4 Performance Evaluation

Fig. 2.4 illustrates for $N = 5$ the monotonicity structure of the optimal block size (Theorem 2.3.1) and verifies that $K_t^* \leq \hat{K}_t$ (Theorem 2.3.2). Both the optimal and greedy block sizes increase when the channel conditions improve (from $\epsilon = 0.5$ to $\epsilon = 0.2$). Next, we evaluate the performance (average system throughput) of different policies. For comparison purposes, we also consider a soft delay-based *conservative* policy, where the scheduler chooses the largest block size with the expected completion time less than or equal to the number of remaining slots. The expected completion time is studied in [23], and it is given by

$$S(K) = K + \sum_{t=K}^{\infty} \left(1 - P(K, t)\right). \tag{2.11}$$

Fig. 2.5 compares the performance of the optimal, greedy, conservative and plain retransmission policies for $N = 10$ and $T = 10$. The plain retransmission policy always performs the worst, whereas the conservative policy performs worse than the greedy policy. However, as ϵ increases, all policies select smaller block sizes and their performance gap diminishes.

Fig. 2.7 shows the tradeoff between the average system throughput and the throughput variation, where $N = 10$ and $T = 20$. When the channels are good (e.g., $\epsilon = 0.1$ in Fig. 2.7), the variation constraint (2.7) makes the scheduler choose a small block size, which reduces the average system throughput accordingly. However, there is no significant effect of (2.7) when channels are bad (e.g., $\epsilon = 0.5$ in Fig. 2.7), since the scheduler already chooses a small block size for large ϵ. Fig. 2.8 evaluates the performance of Algorithm 2 under

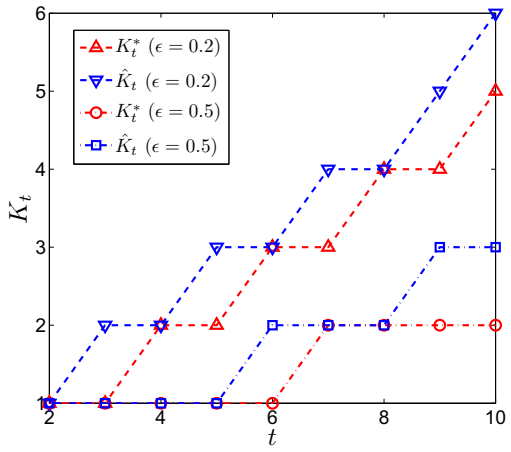

Figure 2.4: K_t^* is nondecreasing and $K_t^* \leq \hat{K}_t$.

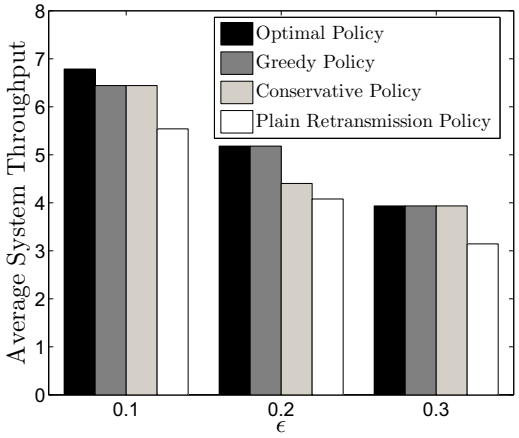

Figure 2.5: Performance (average system throughput) comparison of different policies.

channel uncertainty, where $N = 2$ and $T = 10$ and show that Algorithm 2 is robust with respect to the variation of δ and achieves a reliable throughput performance close to the case with perfect channel information.

The MBIA requires receiver feedback only per packet block (because each receiver sends acknowledgement, only after the block of packets is successfully decoded), compared to plain retransmission policy that requires receiver

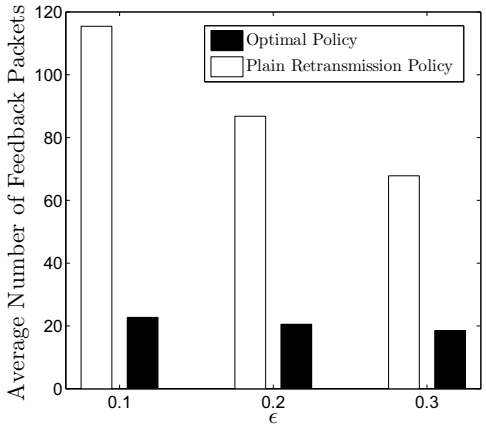

Figure 2.6: Feedback evaluation of different policies.

feedback per packet. Fig. 2.6 shows the reduction in the average number of feedback packets for $N = 10$ and $T = 20$. As ϵ increases, the number of feedback packets decreases, since the probability of decoding each block decreases when the channels become worse.

2.4 Joint Scheduling and Block Size Optimization

In this section, we generalize the study on adaptive NC to the case of multiple frames, where the scheduler serves a set $\mathcal{F}$ of flows subject to the hard deadline and the long-term delivery ratio constraints. The packets of each flow f arrive at the beginning of every frame and they are dropped if they cannot be delivered to its receivers $\mathcal{N}_f$ within this frame (see Fig. 2.1). We impose that the loss probability for flow f due to deadline expiration must be no more than $1 - q_f$, where q_f is the delivery ratio requirement of flow f. For a given frame, the vector $a = (a_f)_{f \in \mathcal{F}}$ denotes the number of packet arrivals at each flow, where a_f is the number of packets generated by flow f. Let x_f be the mean packet arrival rate of flow f. For each flow, we associate a concave

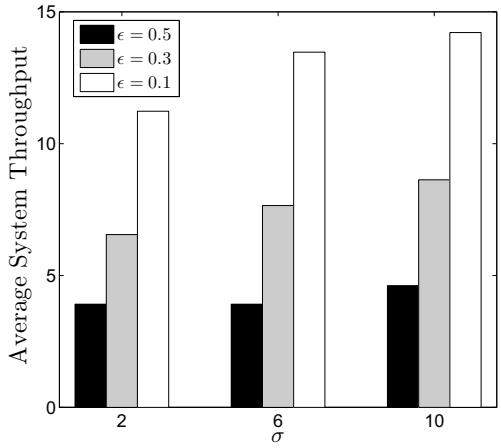

Figure 2.7: Average system throughput vs. throughput variation.

utility function $U_f(x_f)$, based on which the scheduler controls the flow rates to maximize the sum of their utilities subject to the delivery ratio constraints. Due to the hard deadline constraints, the number of packets that can be delivered in each frame is upper-bounded. Accordingly, we denote $X_{\max}$ as the maximum flow rate for each flow. For ease of exposition, we assume perfect channel information at the scheduler and consider coding within each flow but not across different flows.

2.4.1 Multi-flow Scheduling

The scheduler dynamically controls the flow rates, allocates slots for each flow and utilizes the optimal real-time scheduling policy with adaptive NC developed in Section 2.3 to transmit network-coded packets. Our goal is to maximize the system utility subject to the delivery ratio and hard deadline constraints. Due to the random arrivals, the scheduler needs to adjust the flow rates and allocate a suitable number of slots for each flow to satisfy the delivery ratio requirement. Let s_f denote the number of slots allocated to flow f in a frame. This resource allocation is defined as a feasible schedule,

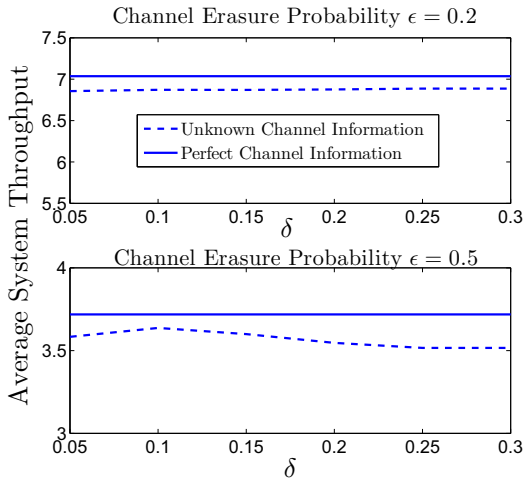

Figure 2.8: Performance of Algorithm 2.

$s = (s_f)_{f \in \mathcal{F}}$, if

$$\sum_{f \in \mathcal{F}} s_f \leq T. \tag{2.12}$$

We intend to find the optimal schedule, i.e., the optimal flow rates and the optimal resource allocation policy, which is defined as the probability $Pr(s|a)$ that given the arrivals a, the schedule $s \in \mathcal{S}$ is chosen from the set $\mathcal{S}$ of all feasible schedules. Then, the expected service rate for flow f is upper-bounded by

$$\mu_f \leq \sum_{s,a} c_f(s_f) Pr(s|a) Pr(a), \tag{2.13}$$

where $Pr(a)$ denotes the probability of the arrivals a, and $c_f(s_f)$ is the expected number of packets that can be delivered under schedule s_f, which is a constant and can be solved by the MBIA. Due to the delivery ratio requirements, we require that

$$\mu_f \geq q_f x_f, \ \forall \ f \in \mathcal{F}. \tag{2.14}$$

Hence, the joint rate control and resource allocation can be formulated

as the following optimization problem:

$$\begin{aligned}
\text{maximize} \quad & \sum_{f\in\mathcal{F}} U_f(x_f) \\
\text{subject to} \quad & \mu_f \geq q_f x_f, 0 \leq x_f \leq X_{\max},\ \forall\ f \in \mathcal{F}, \\
& \mu_f \leq \sum_{s,a} c_f(s_f) Pr(s|a) Pr(a),\ \forall\ f \in \mathcal{F}, \\
& Pr(s|a) \geq 0,\ \forall s \in \mathcal{S},\ \sum_{s\in\mathcal{S}} Pr(s|a) \leq 1,\ \forall a, \\
\text{variables} \quad & \{x_f, \mu_f, Pr(s|a)\}.
\end{aligned} \tag{2.15}$$

Note that (2.15) generalizes the problem studied in [40] by using adaptive NC schemes in packet transmissions and the results in [41] to the joint rate control and resource allocation. Note that the constraints regarding flow rates are coupled with each other forming a rate region due to the underlying slot-based coupling in (2.12).

2.4.2 Dual Decomposition

Since (2.15) is strictly convex, the duality gap is zero from the Slater's condition [42]. The dual problem is given by

$$\begin{aligned}
\text{maximize} \quad & \sum_{f\in\mathcal{F}} (U_f(x_f) + \nu_f(\mu_f - q_f x_f)) \\
\text{subject to} \quad & 0 \leq x_f \leq X_{\max},\ \forall\ f \in \mathcal{F}, \\
& \mu_f \leq \sum_{s,a} c_f(s_f) Pr(s|a) Pr(a),\ \forall\ f \in \mathcal{F}, \\
& Pr(s|a) \geq 0,\ \forall s \in \mathcal{S},\ \sum_{s\in\mathcal{S}} Pr(s|a) \leq 1,\ \forall a, \\
\text{variables} \quad & \{x_f, \mu_f, Pr(s|a)\},
\end{aligned} \tag{2.16}$$

where ν_f is the Lagrangian multiplier associated with the constraint (2.14) for flow f. Note that the problem (2.16) can be decomposed into the following subproblems, namely, the rate control problem and the resource allocation and block size adaptation problem:

$$\max_{0\leq x_f \leq X_{\max}} U_f(x_f) - \nu_f q_f x_f, \tag{2.17}$$

and

$$\begin{aligned} \text{maximize} \quad & \sum_{f\in\mathcal{F}} \nu_f \mu_f \\ \text{subject to} \quad & \mu_f \le \sum_{s,a} c_f(s_f) Pr(s|a) Pr(a),\ \forall\ f \in \mathcal{F}, \\ & Pr(s|a) \ge 0,\ \forall s \in \mathcal{S},\ \sum_{s\in\mathcal{S}} Pr(s|a) \le 1,\ \forall a, \\ \text{variables} \quad & \{\mu_f, Pr(s|a)\}. \end{aligned} \tag{2.18}$$

Since the objective function of (2.18) is concave and the constraints are affine functions, the problem (2.18) can be rewritten as:

$$\max_{s\in\mathcal{S}} \sum_{f\in\mathcal{F}} \nu_f c_f(s_f). \tag{2.19}$$

Thus, we end up with the following iterative algorithm to find the solution to the dual problem (2.16),

$$\begin{aligned} x_f^*(k) &= \arg\max_{0\le x_f\le X_{\max}} U_f(x_f) - \nu_f(k) q_f x_f, \\ s^*(k) &\in \arg\max_{s\in\mathcal{S}} \sum_{f\in\mathcal{F}} \nu_f(k) c_f(s_f), \\ \mu_f^*(k) &= c_f(s_f^*(k)), \\ \nu_f(k+1) &= \max(0, \nu_f(k) + \rho(q_f x_f^*(k) - \mu_f^*(k))), \end{aligned} \tag{2.20}$$

where k is the step index, $\rho > 0$ is a fixed step-size parameter, and $c_f(s_f^*(k))$ is the expected service rate for flow f under schedule $s_f^*(k)$. Letting $\hat{\nu}_f(k) = \frac{\nu_f(k)}{\rho}$, (2.20) is rewritten as

$$\begin{aligned} x_f^*(k) &= \arg\max_{0\le x_f\le X_{\max}} \tfrac{1}{\rho} U_f(x_f) - \hat{\nu}_f(k) q_f x_f, \\ s^*(k) &\in \arg\max_{s\in\mathcal{S}} \sum_{f\in\mathcal{F}} \hat{\nu}_f(k) c_f(s_f), \\ \mu_f^*(k) &= c_f(s_f^*(k)), \\ \hat{\nu}_f(k+1) &= \max(0, \hat{\nu}_f(k) + (q_f x_f^*(k) - \mu_f^*(k))). \end{aligned} \tag{2.21}$$

Remarks: The update equation for $\hat{\nu}_f$ can be interpreted as a virtual queue for the long-term delivery ratio with the arrival rate $q_f x_f^*(k)$ and the

service rate $\mu_f^*(k)$, which keeps track of the deficit in service for flow f to achieve a delivery ratio greater than or equal to q_f. Note that (2.21) provides only the static solution to (2.16). Next, we provide an online scheduling algorithm which takes into account the dynamic arrivals of the flows and the channel realizations.

2.4.3 Online Scheduling Algorithm

The online scheduling algorithm is given by

$$\begin{aligned} x_f^*(k) &= \underset{0\le x_f\le X_{\max}}{\arg\max}\ \tfrac{1}{\rho}U_f(x_f) - \hat{\nu}_f(k)q_f x_f, \\ s^*(k) &\in \underset{s\in\mathcal{S}}{\arg\max} \textstyle\sum_{f\in\mathcal{F}} \hat{\nu}_f(k)c_f(s_f), \\ \hat{\nu}_f(k+1) &= \max\left(0, \hat{\nu}_f(k) + \hat{a}_f(k) - \hat{c}_f(s_f^*(k))\right), \end{aligned} \tag{2.22}$$

where $\hat{c}_f(s_f^*(k))$ denotes the actual number of delivered packets under the schedule $s_f^*(k)$ depending on the channel realizations, and $\hat{a}_f(k)$ is a binomial random variable with parameters $a_f(k)$, the number of packet arrivals of flow f in the kth frame, and q_f. This implementation for $\hat{a}_f(k)$ was proposed in [40]. At the beginning of each period, the schedule $s^*(k)$ is determined by (2.22). Then, the packets of each flow f are transmitted with the MBIA in the scheduled $s_f^*(k)$ slots. The virtual queue $\hat{\nu}_f$ is updated based on the number of successfully delivered packets $\hat{c}_f(s_f^*(k))$ of each flow f. With Lyapunov optimization techniques [43, 44], it can be shown that (2.22) has the following properties.

Theorem 2.4.1. *Consider the Lyapunov function* $L(\hat{\nu}) = \frac{1}{2}\sum_{f\in\mathcal{F}}\hat{\nu}_f^2$. *The expected service deficit* $\hat{\nu}_f$ *is upper-bounded by*

$$\limsup_{k\to\infty} E[\sum_{f\in\mathcal{F}} \hat{\nu}_f(k)] \le B_1 + \tfrac{1}{\rho}B_2, \tag{2.23}$$

for some positive constants B_1 and B_2. Furthermore, the online algorithm can achieve the long-term delivery ratio requirements, i.e., for all $f \in \mathcal{F}$

$$\liminf_{K\to\infty} E[\tfrac{1}{K}\sum_{k=1}^{K} \hat{c}_f(s_f^*(k))] \geq q_f x_f^*, \tag{2.24}$$

where x_f^* is the solution to (2.21).

Proof. The proof is given in Appendix 2.7.7. □

Theorem 2.4.2. *For any $\rho > 0$, the time average system utility under the online algorithm (2.22) is within $B_3\rho$ of the optimal value*

$$\limsup_{K\to\infty} E[\sum_{f\in\mathcal{F}} (U_f(x_f^*) - \tfrac{1}{K}\sum_{k=1}^{K} U_f(x_f^*(k))] \leq B_3\rho, \tag{2.25}$$

where B_3 is a positive constant.

Proof. The proof is given in Appendix 2.7.8. □

Note that the online scheduling algorithm (2.22) can approach within $O(\rho)$ of the optimal solution to (2.16) and does not require any knowledge of the packet arrival statistics.

2.4.4 Performance Evaluation

We consider a network with two flows, each with five receivers. The packet traffic of each flow follows Bernoulli distribution with mean x_f packets/frame for $f = 1, 2$, and the length of each frame is 10 slots. In the simulation, we assume $U_f(x_f) = \log x_f$ for $f = 1, 2$. The channel erasure probabilities for each link are 0.3 in flow 1 and 0.2 in flow 2 respectively.

Fig. 2.9 illustrates how the service deficit of each flow evolves under the online scheduling algorithm (2.22). As shown in Fig. 2.9, the expected service deficit is upper bounded, which corroborates Theorem 2.4.1. Fig. 2.10 depicts

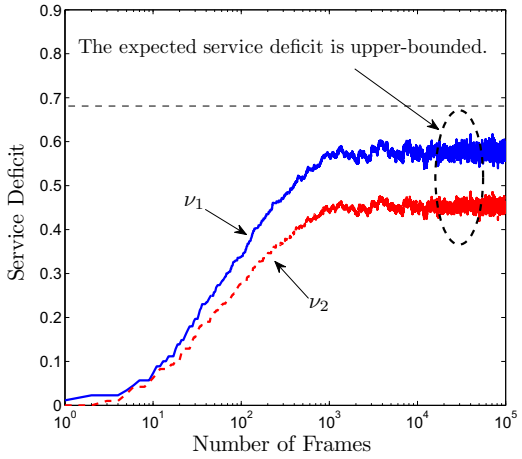

Figure 2.9: Evolution of the service deficit under the online scheduling algorithm.

the performance gap between the solution given by the online scheduling algorithm (2.22) and the static solution given by (2.21) under different step-size ρ. As ρ decreases, the performance gap decreases, which verifies Theorem 2.4.2.

Next, we evaluate the performance of our algorithm by comparing the region of achievable rates (x_1, x_2) with the plain retransmission, where the achievable rates denote the feasible solutions to (2.15) for given delivery ratio requirements q_f. As illustrated in Fig. 2.11, the plain retransmission only achieves a small fraction of the region with adaptive NC. By using adaptive NC, the network can support flows with heavier traffic compared to plain retransmissions.

2.5 High Fidelity Wireless Testing with Hardware Implementation

The adaptive NC schemes are implemented and tested in a realistic wireless emulation environment with real radio transmissions. As illustrated in Fig. 2.12, our testbed platform consists of four main components: radio fre-

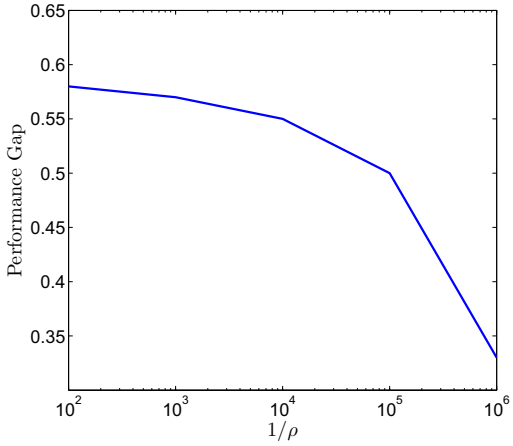

Figure 2.10: Asymptotic optimality of the online scheduling algorithm.

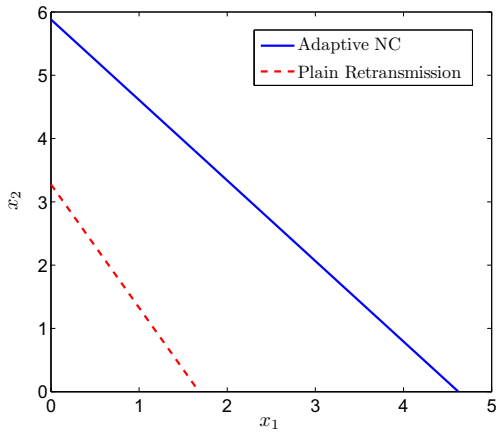

Figure 2.11: Achievable rate regions under adaptive NC and plain retransmission policies.

quency network emulator simulator tool[5], RFnest^{TM} [45], software simulator running higher-layer protocols on a PC host, configurable RF front-ends (RouterStation Pro from Ubiquiti), and digital switch. We removed the radio antennas and connected the radios with RF cables over an attenuator box. Then, real signals are sent over emulated channels, where actual physical-layer

[5]RFnest^{TM}is developed and owned as a trademark by Intelligent Automation, Inc.

interactions occur between radios, and in the meantime the physical channel attenuation is digitally controlled according to the simulation model or recorded field test scenarios can be replayed accordingly.

In the hardware experiments, wireless tests are executed at 2.462GHz channel with 10dBm transmission power and 1Mbps rate. CORE (Common Open Research Emulator) [46] is used to manage the scenario being tested. The locations of receivers are changed through RFnest^{TM}GUI, where the signal power is modeled to decay as $d^{-\alpha}$ over distance d with path loss coefficient $\alpha = 4$. By using real radio transmissions according to this model, we varied the attenuation from the transmitter to each of the receivers and generated different channel erasure probabilities. With RFnest^{TM}, the same wireless traces for each of the NC algorithms are replayed and compared under the high fidelity network emulation with hardware-in-the-loop experiments.

Fig. 2.13 illustrates the performance of the optimal policy, the greedy policy and the fixed block size policy suggested by [36]. The experimental results show that the greedy policy performs close to the optimal policy in practice. Both the greedy and the optimal policies outperform the fixed block size policy, and the complexity remains low with the polynomial-time algorithm MBIA. Fig. 2.14 illustrates the wireless test performance for the case when the unknown channel erasure probabilities are learned over time. Algorithm 2 performs close to optimal in this case and converges quickly in several frames.

2.6 Conclusion

The adaptive NC for multimedia traffic with hard deadlines was considered and formulated as a Markov decision process for a single-hop wireless network. By

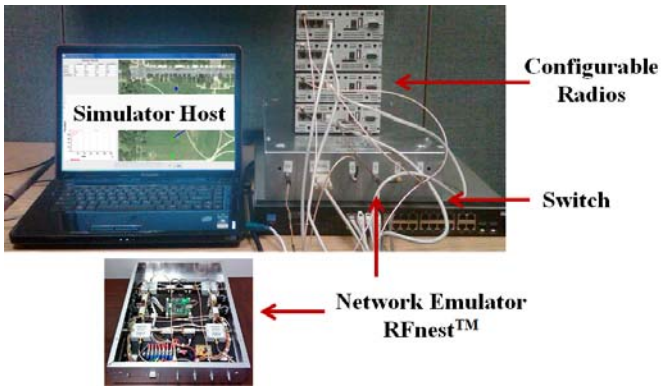

Figure 2.12: Programmable RFnest^{TM} testbed.

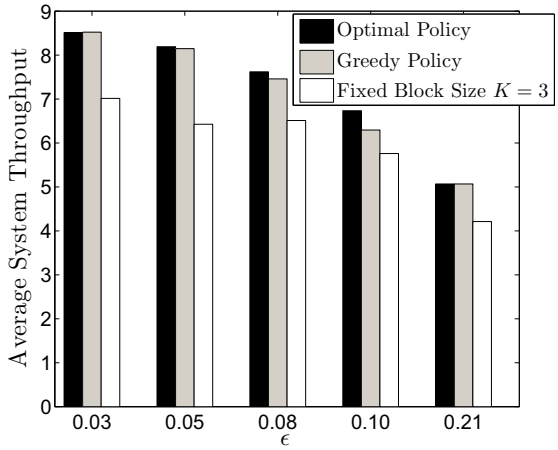

Figure 2.13: Performance comparison of different NC block size adaptation policies with network emulation, where $N = 10$ and $T = 10$.

exploring the structural properties of the problem, the polynomial time policy, MBIA, was derived to solve the optimal NC block size adaptation problem and the joint real-time scheduling and channel learning scheme was developed to adapt MBIA to wireless channel dynamics for cases when the perfect channel information is not necessarily available at the scheduler. Then, the study was generalized to multiple flows with hard deadlines and long-term delivery constraints, and a low-complexity online scheduling algorithm was integrated

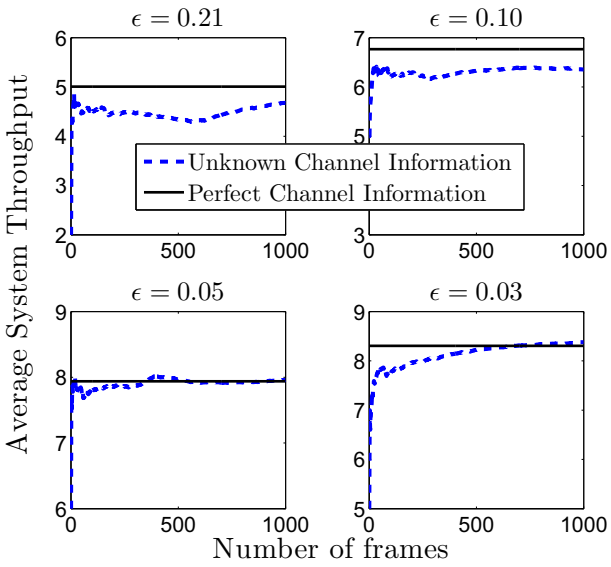

Figure 2.14: Average system throughput and convergence rate of Algorithm 2, where $N = 10$, $T = 10$ and the initial channel erasure probability estimation is 0.5.

with the MBIA to swerve multiple flows over multiple frames. Finally, high fidelity wireless emulation tests with real radios were performed to demonstrate the feasibility of the MBIA in finding the optimal block size in real time.

2.7 Appendix

2.7.1 Proof of Lemma 2.2.1

$P(K,T)$ is monotonically decreasing with K, if $\hat{P}(K,T)$ is monotonically decreases with K, where

$$\hat{P}(K,T) = \sum_{\tau=K}^{T} \binom{\tau-1}{K-1} \epsilon^{\tau-K}(1-\epsilon)^K. \tag{2.26}$$

such that $P(K,T) = (\hat{P}(K,T))^N$. First, we express

$$\hat{P}(K,T) - \epsilon\hat{P}(K,T)$$

$$= (1-\epsilon)^K + \sum_{\tau=K+1}^{T} \left(\binom{\tau-1}{K-1} - \binom{\tau-2}{K-1} \right) \epsilon^{\tau-K}(1-\epsilon)^K - \binom{T-1}{K-1}\epsilon^{T-(K-1)}(1-\epsilon)^K$$

$$= (1-\epsilon)^K + \sum_{\tau=K+1}^{T} \binom{\tau-2}{K-2} \epsilon^{\tau-K}(1-\epsilon)^K - \binom{T-1}{K-1}\epsilon^{T-(K-1)}(1-\epsilon)^K$$

$$= (1-\epsilon)\left(\sum_{\tau=K}^{T}\tbinom{\tau-2}{K-2}\epsilon^{\tau-K}(1-\epsilon)^{K-1}\right) - \tbinom{T-1}{K-1}\epsilon^{T-(K-1)}(1-\epsilon)^{K}$$

$$= (1-\epsilon)\Big(\sum_{\tau=K-1}^{T}\tbinom{\tau-1}{K-2}\epsilon^{\tau+1-K}(1-\epsilon)^{K-1} - \tbinom{T-1}{K-2}\epsilon^{T+1-K}(1-\epsilon)^{K-1}\Big)$$
$$- \tbinom{T-1}{K-1}\epsilon^{T-(K-1)}(1-\epsilon)^{K}$$

$$= (1-\epsilon)\hat{P}(K-1,T) - \tbinom{T-1}{K-2}\epsilon^{T+1-K}(1-\epsilon)^{K} - \tbinom{T-1}{K-1}\epsilon^{T-(K-1)}(1-\epsilon)^{K}$$

$$= (1-\epsilon)\hat{P}(K-1,T) - \tbinom{T}{K-1}\epsilon^{T-(K-1)}(1-\epsilon)^{K}. \tag{2.27}$$

Since from (2.27) it follows that

$$\hat{P}(K,T) - \hat{P}(K-1,T) = -\tbinom{T}{K-1}\epsilon^{T-(K-1)}(1-\epsilon)^{K-1} \tag{2.28}$$

is negative, $P(K,T)$ is monotonically decreasing with K.

2.7.2 Proof of Proposition 2.3.1

1) To show that $R_t(K_t)$ is unimodal, it suffices to show that $R_t(K_t)$ is log-concave, i.e., $\hat{R}(K) = \frac{1}{N}\log(K) + \log(\hat{P}(K,T))$ is concave. Since $\frac{1}{N}\log(K)$ is concave, it suffices to show that for any given T, $\log(\hat{P}(K,T))$ is concave, i.e., $\hat{P}(K,T)$ is log-concave. Based on the definition of log-concavity, in what follows, we will show that

$$\hat{P}(K,T)^2 \geq \hat{P}(K-1,T)\hat{P}(K+1,T). \tag{2.29}$$

Based on (2.28), (2.29) can be rewritten as

$$\hat{P}(K-1,T)\left(\tfrac{T-K+1}{K}\right)\tfrac{1-\epsilon}{\epsilon} - \hat{P}(K,T) \geq 0. \tag{2.30}$$

We use induction to show (2.30). For $T = 1, 2$, it is obvious to see that (2.30) holds. For $T = 3$, we can verify (2.30) by using (2.26). Assume that for

$T = t > 3$, (2.30) holds. For $T = t + 1$, after some algebra, we have

$$\begin{aligned} & \hat{P}(K-1,t+1)(\tfrac{t-K+2}{K})\tfrac{1-\epsilon}{\epsilon} - \hat{P}(K,t+1) \\ = \; & \hat{P}(K-1,t)(\tfrac{t-K+1}{K})\tfrac{1-\epsilon}{\epsilon} - \hat{P}(K,t) \\ & + \tfrac{1}{K}(\hat{P}(K-1,t)\tfrac{1-\epsilon}{\epsilon} - \tbinom{t}{K-1}\epsilon^{t-(K-1)}(1-\epsilon)^K) \\ \geq \; & 0, \end{aligned} \tag{2.31}$$

which is based on the induction and (2.28).

2) The proof follows from a contradiction argument. Suppose that $\hat{K}_t > \hat{K}_{t+1}$. Since $\hat{K}_t$ is the greedy block size in slot t, we have $R_t(\hat{K}_t) \geq R_t(\hat{K}_{t+1})$. Since $P(K_t, t)$ is monotonically increasing with t for any K_t, $R_t(K_t)$ is monotonically increasing with t. It can be shown that $R_{t+1}(K_1) - R_t(K_1) > R_{t+1}(K_2) - R_t(K_2)$ for any $0 < K_2 < K_1 \leq \hat{K}_t$. Therefore, when $\hat{K}_t > \hat{K}_{t+1}$, we have $R_{t+1}(\hat{K}_t) \geq R_{t+1}(\hat{K}_{t+1})$, which contradicts the fact that $\hat{K}_{t+1}$ is the greedy block size in slot $t+1$. Therefore, $\hat{K}_{t+1} \geq \hat{K}_t$.

2.7.3 Proof of Theorem 2.3.1

The proof follows from a contradiction argument. Suppose that $K_t^* > K_{t+1}^*$. It can be shown that $K_t^* \leq \hat{K}_t$ by following the similar steps in Appendix 2.7.4. From Proposition 2.3.1, it follows that $R_t(K_t^*) > R_t(K_{t+1}^*)$ in slot t. Since K_t^* is the optimal action in slot t, $V_t(K_t^*; \mathcal{P}^*) > V_t(K_{t+1}^*; \mathcal{P}^*)$. Since $K_t^* > K_{t+1}^*$ in slot t, when the optimal policy is applied, the future reward $J_t(K_t^*)$ under K_t^* is less than the future reward $J_t(K_{t+1}^*)$ under K_{t+1}^*, due to the less remaining time under K_t^*. Since $R_t(K)$ is monotonically increasing with t for given K, $R_{t+1}(K_t^*) > R_{t+1}(K_{t+1}^*)$ in slot $t+1$. Since K_{t+1}^* is the optimal action in slot $t+1$, $V_{t+1}(K_t^*; \mathcal{P}^*) < V_{t+1}(K_{t+1}^*; \mathcal{P}^*)$. Therefore, $J_{t+1}(K_t^*)$ is less than $J_{t+1}(K_{t+1}^*)$. Note that $J_t(K_t^*) - J_t(K_{t+1}^*)$ decreases monotonically as t increases, i.e., the gap between these future rewards decreases in slot

$t+1$. Hence we have $R_{t+1}(K^*_{t+1}) - R_t(K^*_{t+1}) > R_{t+1}(K^*_t) - R_t(K^*_t)$, which contradicts the fact that $R_{t+1}(K_1) - R_t(K_1) > R_{t+1}(K_2) - R_t(K_2)$ for any $0 < K_2 < K_1 \leq \hat{K}_t$ based on Proposition 2.3.1. Thus, Theorem 2.3.1 follows.

2.7.4 Proof of Theorem 2.3.2

The proof follows from a contradiction argument. Suppose that $K^*_t > \hat{K}_t$. For any sample path, the case with $\hat{K}_t$ will deliver the block earlier than the case with K^*_t. For the sample paths with the number of slots that all the channels between the transmitter and the receivers are good less than K^*_t, the reward under $\hat{K}_t$ is higher than that under K^*_t, which equals to zero. For the other sample paths with the number of slots that all the channels between the transmitter and the receivers are good greater than K^*_t, the block with size $\hat{K}_t$ will be delivered earlier than that with size K^*_t. We assume that after the block with size $\hat{K}_t$ is delivered, the scheduler chooses to wait until the block with size K^*_t is delivered. Then after the block with size K^*_t is delivered, the optimal policy is applied for both cases. Obviously, in this case, both cases with K^*_t and $\hat{K}_t$ will generate the same future reward. However, based on the definition of greedy block size, the average immediate reward with block size $\hat{K}_t$ is larger than that with block size K^*_t. Therefore, taking into account all the sample paths, the total reward with block size $\hat{K}_t$ is larger than that with block size K^*_t, which contradicts the fact that K^*_t is the optimal action in slot t. Therefore, Theorem 2.3.2 follows.

2.7.5 Proof of Theorem 2.3.3

Based on Proposition 2.3.1, $R_t(K_t)$ has the *unimodal* property and therefore $\hat{K}_t$ can be solved efficiently by the Fibonacci search algorithm [47], which is a sequential line search algorithm with a complexity of $O(\log(t))$ at state t.

Therefore, in each iteration, it takes $O(\log(t) + \hat{K}_t - K^*_{t-1})$ slots to find K^*_t. Based on Lemma 2.3.1, $\hat{K}_t - K^*_{t-1}$ is upper bounded by t. After some algebra, we show that the complexity of Algorithm 1 is bounded by $O(T^2)$ and Theorem 2.3.3 follows.

2.7.6 Proof of Theorem 2.3.4

When $R_t(1) > R_t(2)$ holds, $\hat{K}_t = 1$, due to the unimodal property of $R_t(\cdot)$. Then, $K^*_t = 1$ from Theorem 2.3.2. Since K^*_t is non-decreasing with t (Theorem 2.3.1), $K^*_{t'} = 1$ in the remaining slots $t' > t$, i.e., the plain retransmission policy is optimal. To show there exits a threshold ϵ^*, we expand $R_t(1) > R_t(2)$ according to (2.2), where $R_t(1) = (1-\epsilon^T)^N$ and $R_t(2) = 2(1-\epsilon^T + T\epsilon^T - T\epsilon^{T-1})^N$. Then, the monotonicity of ϵ^* follows from comparing $R_t(1)$ with $R_t(2)$ in the expanded form. Define $f(\epsilon, t, N) = (1-\epsilon^t) - 2^{1/N}(1-\epsilon^t + t\epsilon^t - t\epsilon^{t-1})$ such that $f(\epsilon^*(t,N), t, N) = 0$. Note that $f(0,t,N) = 1 - 2^{1/N} < 0$ and $f(1,t,N) = 0$. There exists a unique non-trivial value of ϵ' in $(0,1)$ to maximize $f(\epsilon, t, N)$. For $\epsilon < \epsilon'$, $f(\epsilon, t, N)$ is first increasing and then decreasing back to 0. Therefore, there exists a unique non-trivial solution of $f(\epsilon^*(t,N), t, N) = 0$ such that $f(\epsilon, t, N) < 0$ for $\epsilon < \epsilon^*(t,N)$ and $f(\epsilon, t, N) > 0$ for $\epsilon > \epsilon^*(t,N)$.

2.7.7 Proof of Theorem 2.4.1

For the purpose of the proof, we define the capacity region for fixed arrivals as

$$\mathcal{C}(a) = \left\{ (\mu_f(a))_{f\in\mathcal{F}} \middle| \begin{array}{l} \mu_f(a) \le \sum\limits_{s,a} c_f(s_f) Pr(s|a), \forall f \in \mathcal{F} \\ \sum\limits_{s\in\mathcal{S}} Pr(s|a) \le 1, Pr(s|a) \ge 0 \end{array} \right\}.$$

Accordingly, the overall capacity region is defined as

$$\mathcal{C} = \{(\mu_f)_{f\in\mathcal{F}} | \mu_f = E[\mu_f(a)], \forall f \in \mathcal{F}\}.$$

Define the conditional Lyapunov drift $\Delta(\hat{\nu}(k))$ as follows:

$$\Delta(\hat{\nu}(k)) = E[L(\hat{\nu}(k+1)) - L(\hat{\nu}(k))|\hat{\nu}(k)]. \tag{2.32}$$

We expand $\Delta(\hat{\nu}(k))$ by using $\hat{\nu}_f(k+1) = \max(0, \hat{\nu}_f(k) + \hat{a}_f(k) - \hat{c}_f(s_f^*(k)))$ given in (2.22) as follows:

$$\begin{aligned}
& \Delta(\hat{\nu}(k)) \\
= \; & E\Big[\tfrac{1}{2}\sum_f \big(\max(0, \hat{\nu}_f(k) + \hat{a}_f(k) - \hat{c}_f(s_f^*(k)))\big)^2 - \tfrac{1}{2}\sum_f \nu_f^2(k)|\nu(k)\Big] \\
\leq \; & E\Big[\tfrac{1}{2}\sum_f \big(\hat{\nu}_f(k) + \hat{a}_f(k) - \hat{c}_f(s_f^*(k))\big)^2 - \tfrac{1}{2}\sum_f \nu_f^2(k)|\nu(k)\Big] \\
= \; & E\Big[\sum_f \nu_f(k)(\hat{a}_f(k) - \hat{c}_f(s_f^*(k))) + \tfrac{1}{2}\sum_f \big(\hat{a}_f(k) - \hat{c}_f(s_f^*(k))\big)^2 |\nu(k)\Big] \\
\leq \; & E\Big[\sum_f \nu_f(k)(\hat{a}_f(k) - \hat{c}_f(s_f^*(k))) + \tfrac{1}{2}\sum_f \big(\hat{a}_f(k)^2 + \hat{c}_f(s_f^*(k))^2\big) |\nu(k)\Big] \\
\overset{(a)}{\leq} \; & B_3 + \sum_f \nu_f(k)\big(q_f x_f^*(k) - E[\hat{c}_f(s_f^*(k))|\nu(k)]\big) \\
= \; & B_3 - \sum_f \Big(\tfrac{1}{\rho}U_f(x_f^*(k)) - \nu_f(k) q_f x_f^*(k)\Big) \\
& + \sum_f \tfrac{1}{\rho}U_f(x_f^*(k)) - \sum_f \nu_f(k) E[\hat{c}_f(s_f^*(k))|\nu(k)] \\
\overset{(b)}{\leq} \; & B_3 - \sum_f \Big(\tfrac{1}{\rho}U_f(x_f^*) - \nu_f(k) q_f x_f^*\Big) \\
& + \sum_f \tfrac{1}{\rho}U_f(x_f^*(k)) - \sum_f \nu_f(k) E[\hat{c}_f(s_f^*(k))|\nu(k)] \\
= \; & B_3 - \tfrac{1}{\rho}\sum_f \big(U_f(x_f^*) - U_f(x_f^*(k))\big) \\
& - \sum_f \nu_f(k)\big(E[\hat{c}_f(s_f^*(k))|\nu(k)] - q_f x_f^*\big) \\
\overset{(c)}{\leq} \; & B_3 - \tfrac{1}{\rho}\sum_f \big(U_f(x_f^*) - U_f(x_f^*(k))\big) - \sum_f \nu_f(k)\big(E[\mu_f(a)] - q_f x_f^*\big) \\
\overset{(d)}{=} \; & B_3 - \tfrac{1}{\rho}\sum_f \big(U_f(x_f^*) - U_f(x_f^*(k))\big) - \sum_f \nu_f(k)\big(\mu_f - q_f x_f^*\big)
\end{aligned}$$

where (a) B_3 is a positive constant such that

$$B_3 \geq E[\frac{1}{2}\sum_f (\hat{a}_f(k)^2 + \hat{c}_f(s_f^*(k))^2)|\nu(k)]$$

due to the fact that both $\hat{a}_f(k)$ and $\hat{c}_f(s_f^*(k)$ are upper bounded; (b) follows from the fact that $x_f^*(k)$ is the optimal solution to (2.22); (c) follows for any

$(\mu_f(a))_{f\in\mathcal{F}} \in \mathcal{C}(a)$; (d) follows for any $(\mu_f)_{f\in\mathcal{F}} \in \mathcal{C}$. Since $(\mu_f^*)_{f\in\mathcal{F}} \in \mathcal{C}$ is the solution to (2.21), we have

$$\Delta(\hat{\nu}(k)) \leq B_3 - \tfrac{1}{\rho}\sum_f \big(U_f(x_f^*) - U_f(x_f^*(k))\big) - B_4 \sum_f \nu_f(k), \tag{2.33}$$

where $B_4 = \min_{f\in\mathcal{F}}\{\mu_f^* - q_f x_f^*\}$ is a positive number.

Let $B_5 = \sum_f \max_{0\leq x_l \leq X_{\max}} U_l(x_l)$. Then we have

$$\Delta(\hat{\nu}(k)) \leq B_3 + \tfrac{1}{\rho}B_5 - B_4 \sum_f \nu_f(k). \tag{2.34}$$

Therefore, $\Delta(\hat{\nu}(k))$ is bounded. Hence, (2.23) follows, where $B_1 = B_3/B_4$ and $B_2 = B_5/B_4$.

The proof of (2.24) is based on the stability of the deficit queues. Note that the cumulative deficit to fulfill the QoS constraint is bounded by the deficit queue,

$$\sum_{k=1}^{K} \big(\hat{a}_f(k) - \hat{c}_f(s_f^*(k))\big) \leq \hat{v}_f(K). \tag{2.35}$$

Since $\hat{v}_f(K)$ is bounded, $\hat{v}_f(K)/K \to 0$, as $K \to \infty$. Therefore, the QoS constraints are met, i.e., (2.24) follows.

2.7.8 Proof of Theorem 2.4.2

Taking the expectations of (2.33) and manipulating the terms, we obtain

$$\begin{aligned} & E\left[\tfrac{1}{\rho}\sum_f \big(U_f(x_f^*) - U_f(x_f^*(k))\big)\right] \\ \leq\ & B_3 - E[L(\nu(k+1))] + E[L(\nu(k))] - B_4 \sum_{f\in\mathcal{F}} E[\nu_f(k)] \\ \leq\ & B_3 - E[L(\nu(k+1))] + E[L(\nu(k))]. \end{aligned}$$

Summing over $k = \{1, \cdots, K\}$ and dividing by K, we have,

$$\begin{aligned} & E\left[\tfrac{1}{\rho}\sum_f \left(U_f(x_f^*) - \tfrac{1}{K}\sum_{k=1}^{K} U_f(x_f^*(k))\right)\right] \\ \leq \;& B_3 - \tfrac{E[L(\nu(K+1))]}{K} + \tfrac{E[L(\nu(1))]}{K} \\ \leq \;& B_3 + \tfrac{E[L(\nu(1))]}{K}. \end{aligned}$$

Since $E[L(\nu(1))] \leq \infty$, taking a lim sup yields (2.25).

Chapter 3

DOWNLINK SCHEDULING FOR REAL-TIME TRAFFIC OVER MARKOVIAN CHANNELS

3.1 Introduction

3.1.1 Motivation

Recent years have witnessed a tremendous growth in multimedia applications in wireless systems [48]. Real-time applications such as VoIP and real-time video will continue to take communications to the next level beyond texts and images. Typically, real-time traffic has more stringent quality of service (QoS) requirements, in terms of minimum bandwidth and maximum packet delay. Besides, real-time traffic is prioritized at the packet level, e.g., in video streaming, video packets encoded by MPEG, have different priorities according to the distortion reduction of each packet in reconstructing the original media [49]. All of these features make the real-time scheduling problem challenging, not to mention the unreliable nature of wireless channels.

A large number of wireless scheduling algorithms were developed for the case without deadline constraints (see, e.g., [50–52] and the references therein), and there have been only very limited attempts on real-time scheduling in wireless networks (see, e.g., [38, 53–56]). Recent works [38, 53, 54] studied admission control and scheduling for inelastic flows, assuming a two-state *i.i.d.* flat fading channel. Due to its simplicity, the *i.i.d.* model has been a popular choice used to abstract the fading channels. It is clear, however, this model fails to capture the time-correlation or the channel memory – an important characteristic of realistic wireless channels [2]. Notably, [55, 56] used a Markovian channel model to capture the channel memory and studied scheduling

under strict packet deadlines accordingly, assuming that perfect channel state information (CSI) is readily available at the scheduler. In practical scenarios, however, perfect CSI is rarely available and never cost-free, i.e., a non-trivial amount of network resources, that could otherwise be used for data transmissions, must be spent in learning the channel conditions. Joint channel learning and scheduling is therefore the need of the hour. Worth noting is that more recent works [57–59] investigated scheduling for "best-effort" services under Markovian channel models while exploiting the channel memory to perform joint channel estimation and scheduling. Specifically, [57] considered opportunistic spectrum access in cognitive radios with Markovian channels and 1-bit feedback, whereas [58] studied exploiting channel memory for scheduling in a downlink system using randomly delayed ARQ feedback.

3.1.2 Summary of Main Results

In this chapter, we study downlink scheduling of multiuser traffic with packet-level priorities and hard deadlines. The priority of real-time traffic is modeled by assigning each packet a weight to indicate its importance. The user channels are modeled as Markovian, and the base station can learn, via feedback, only the channel condition of the currently scheduled user. The feedback thus obtained will be used to predict the channels for future scheduling decisions. That is to say, channel learning is tightly coupled with the scheduling problem. We cast this joint channel learning and scheduling problem as a partially observable Markov decision process [60], while taking into account the strict packet deadlines.

We show that, when the strict deadline constraint of real-time traffic meets the partial observability of the channels, the scheduling problem involves a more intrinsic and complicated tradeoff beyond the classic "exploitation

vs exploration". Indeed, we show that the "exploitation and exploration" tradeoffs take place on two timescales, i.e., one at the slot level and the other at the period level associated with the packet deadlines (to be made more precise in Section 3.3). Specifically, at the slot level, the tradeoff has a classic flavor, whereas, at the period level, the tradeoff is tied with the successful transmission of packets backlogged in the current period and that of arrivals in the future periods. The period-level tradeoff is characterized by the relevance of the *idling* action, where the scheduler does not transmit any packet, with the intention to improve the "freshness" of channel state information for the future periods. Through a case study, we show that idling is indeed an optimal action under certain system states in the light traffic regime. In contrast, for the special case with a symmetric two-user system, it turns out that idling is suboptimal. In fact, based on rather involved analysis, we discover that a non-idling greedy policy is optimal for this special case.

These somewhat surprising results involving the idling action reveal the existence of the fundamental tradeoffs between exploitation and exploration/idling. Essentially, it is the unique dynamics tied with packet deadlines, priorities and channel memory that presents unique challenges beyond the models studied in [57–59]. We believe that these initial steps we have taken here open a new avenue for real-time scheduling in wireless networks, and shed light on QoS provisioning therein.

The rest of the chapter is organized as follows. In Section 3.2, we introduce the system model and describe the problem formulation. We then discuss in Section 3.3 the optimal scheduling policy and the fundamental tradeoff inherent in the scheduling problem, followed by a case study on the optimality of idling decision in Section 3.4. Then we study in Section 3.5 the optimal

policy in the symmetric two-user system, followed by concluding remarks in Section 3.6.

3.2 Problem Formulation

3.2.1 Channel Model

We consider a time-slotted downlink system with one base station and N users, with time slots synchronized across users. Each user is associated with a real-time packet traffic flow that generates packets with different priority levels (see, e.g., the packet-level priority model in MPEG video streaming [49]). We will elaborate further on the priority model in the next subsection. The channel between the base station and each user is modeled by a two-state Markov chain. Each state corresponds to the degree of decodability of the packet sent through the channel, with state 1 (ON) corresponding to full decodability and state 0 (OFF) to non-decodability. The Markov channels are assumed to be independent but, in general, non-identical across users. For each user, the channel remains fixed in a slot and evolves to another state in the next slot following the state transition probabilities of the Markov chain. The 2×2 probability transition matrix of the Markov channel of user i is given by

$$\mathbf{P_i} = \begin{bmatrix} p_i & 1-p_i \\ r_i & 1-r_i \end{bmatrix}, \tag{3.1}$$

where p_i denotes the probability that the channel remains in the ON state as the system evolves across adjacent slots, while r_i denotes the probability of transition from OFF state to ON state. Note that, the Markov channel states can be interpreted as a quantized representation of the underlying channel strength. In practical scenarios, a discretized fading channel is more likely to stay in the current state than to crossover to another state [61], i.e., $p_i > r_i$.

We adopt this condition in what follows.

3.2.2 Scheduling Real-time Traffic with Priorities

As is standard [38, 53, 54], time is divided into periods (frames), with each period further divided into equal number of time slots. We assume that there are K periods till the end of the scheduling process and T slots per period. Besides, we consider the case that at the beginning of each period τ, each real-time packet traffic flow i generates a new packet with priority, $q_\tau(i)$. The priorities, $q_\tau(i)$, are *i.i.d.* across users and periods, and follow a distribution $\mathbf{Q}$. Suppose that these packets must be delivered to their respective users before the end of the period, i.e., the packet deadline is given by the length of the period, T slots. Any packet not delivered by this deadline would be dropped from the system, as illustrated in Fig. 3.1. Also assume that the transmission time of a packet corresponds to one time slot.

In each time slot, the base station (scheduler) can schedule no more than one user with a packet yet to be transmitted. If a user is scheduled, upon the successful reception of the corresponding packet, the scheduled user sends back an ACK to the scheduler at the end of the slot, over an error-free feedback channel, while a failed packet reception triggers a NACK feedback. This feedback is then used by the scheduler for future scheduling decisions (to be elaborated in the sequel). Also, motivated by energy consideration in *mobile* wireless systems, we assume that the base station cannot transmit a *dummy* packet to a user to learn the channel condition via ARQ feedback.

3.2.3 Problem Formulation: A POMDP View

The real-time traffic scheduling problem above is a sequential decision problem, and can be treated as a partially observable Markov decision process. We now

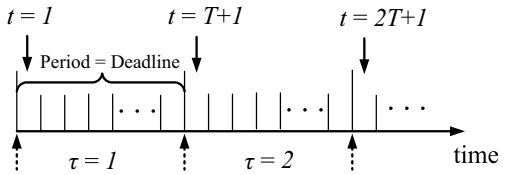

Figure 3.1: A sketch of the slot and period structure. (The dotted arrow denotes the time instant for both new packet arrivals and undelivered packet drops.)

elaborate on the various entities involved, in the language of POMDP.

Horizon: The number of consecutive periods over which the scheduling decisions are made is the horizon. In this chapter, we focus on the finite horizon problem with K periods, or equivalently, with KT slots.

Belief value: Let $\pi_t(i) \in [0,1]$ be the belief value of the ith user in slot t, defined as the probability that the ith user's channel is in ON state in the current slot. The belief value of user i evolves from the current slot to the next slot as follows: if user i is scheduled in the current slot, then $\pi_{t+1}(i) = p_i$ if an ACK is received at the end of the slot and $\pi_{t+1}(i) = r_i$ if a NACK is received. If user i is not the current scheduling choice, then $\pi_{t+1}(i) = \mathcal{T}(\pi_t(i))$, where $\mathcal{T} : [0,1] \mapsto [0,1]$ is the belief evolution operator given by

$$\mathcal{T}(\pi_t(i)) = \pi_t(i)p_i + (1 - \pi_t(i))\, r_i. \tag{3.2}$$

Action: Let $a_t \in \{0, 1, 2, \cdots, N\}$ indicate the action taken in slot t, where $a_t = i$, $i \in \{1, \cdots, N\}$ corresponds to scheduling the ith user in slot t. If $a_t = 0$, the scheduler does not serve any user, i.e., it stays *idle*. We will discuss the significance of including the idle action in the action space in the next section.

Expected immediate reward: In slot t, if user i is scheduled, the sched-

uler accrues a reward of $q_{\tau(t)}(i)$ if and only if this user has an undelivered packet and the packet is successfully delivered. Note that $\tau(t)$ denotes the period containing the slot t. The expected immediate reward accrued by the scheduler, as a function of the belief values and action, is given by

$$R_t(\pi_t, \mathcal{S}_t, q_{\tau(t)}; a_t) = \begin{cases} q_{\tau(t)}(a_t)\pi_t(a_t), & a_t \in \mathcal{S}_t \\ 0, & \text{otherwise} \end{cases}, \tag{3.3}$$

where $\mathcal{S}_t$ is the set of users with an undelivered packet in the current slot t.

Scheduling policy: A scheduling policy $\mathcal{P}$ is a sequence of mappings, $\mathcal{P} = \{\mathcal{P}_t\}_{t=1}^{KT}$, from π_t, $\mathcal{S}_t$, and $q_{\tau(t)}$ to an action $a_t \in \{0, 1, 2, \cdots, N\}$ in each slot $t \in \{1, \ldots, KT\}$, i.e., $a_t = \mathcal{P}_t(\pi_t, \mathcal{S}_t, q_{\tau(t)})$.

Total discounted reward: Given the scheduling policy $\mathcal{P}$, the total discounted reward in slot t is given by

$$V_t(\pi_t, \mathcal{S}_t, q_{\tau(t)}; \mathcal{P}) = R_t(\pi_t, \mathcal{S}_t, q_{\tau(t)}; a_t) + \beta E[V_{t+1}(\pi_{t+1}, \mathcal{S}_{t+1}, q_{\tau(t+1)}; \mathcal{P})], \tag{3.4}$$

where the first quantity is the expected immediate reward, and the second quantity is the expected discounted future reward. The expectation in the second quantity takes into account the stochastic of the Markov channel states and the packet priorities in the future. The discount factor $\beta \in [0, 1]$ relatively weighs the future reward with respect to the immediate reward.

3.3 Slot-level and Period-level Tradeoffs

3.3.1 Optimal Scheduling Policy

A scheduling policy is optimal if and only if it achieves the optimal reward given by the Bellman equation [39]:

$$V_t(\pi_t, \mathcal{S}_t, q_{\tau(t)}) = \max_{a_t \in \{0,1,\cdots,N\}} \{R_t(\pi_t, \mathcal{S}_t, q_{\tau(t)}; a_t) + \beta E[V_{t+1}(\pi_{t+1}, \mathcal{S}_{t+1}, q_{\tau(t+1)})]\}, \tag{3.5}$$

with the terminal reward

$$V_{KT}(\pi_{KT}, \mathcal{S}_{KT}, q_{\tau(KT)}) = \max_{a_{KT} \in \{0,1,\cdots,N\}} R_{KT}(\pi_{KT}, \mathcal{S}_{KT}, q_{\tau(KT)}; a_{KT}).$$

We now proceed to discuss the fundamental tradeoff inherent in the scheduling problem herein.

3.3.2 Fundamental Tradeoffs

In each slot, the scheduling decision must take into account 1) immediate gains through data transmission in the current slot and 2) future gains, since the current decision influences the future trajectory of CSI at the scheduler. This is the "exploitation vs exploration" tradeoff traditionally found in sequential decision problems (e.g., [62]). The "exploitation" side of the tradeoff is favored by scheduling the user with the best perceived channel condition at the moment, and this "disregards" the future trajectory of the scheduling process. On the other hand, *traditionally* in a backlogged system, the "exploration" side of the tradeoff is favored, in a broad sense, by scheduling the user whose CSI at the scheduler carries the most uncertainty.

Next, we take a closer look at the dynamics of the problem under consideration. Due to the finiteness of the packets available in each period, it is

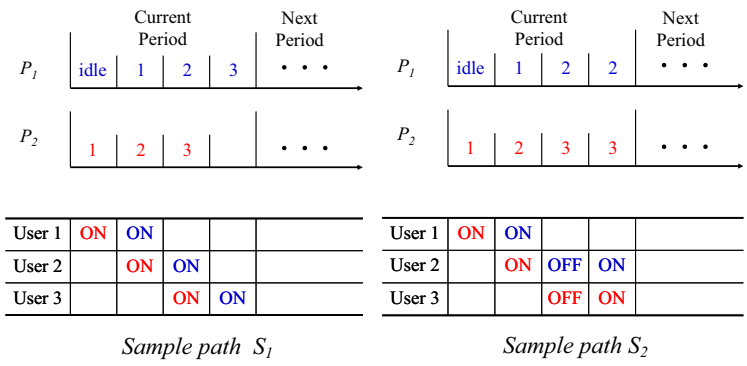

Figure 3.2: Idling in the context of the period-level tradeoff: the optimal scheduling policy is not necessarily non-idling.

possible that the scheduler completes successful transmission of all the packets ahead of the end of the period and stays idle (i.e., not collect feedback) for the rest of the period. Clearly, the last slot in which a packet was transmitted is the last slot in which a feedback was received. Roughly speaking, the closer this *last feedback slot* is to the end of the period, the "more fresh" is the CSI available at the scheduler, for the next period. Thus there appears to be an incentive to deliberately delay transmissions in a period. One natural way to achieve this is through idling, i.e., not schedule any user, in appropriately identified slots. On the other hand, the idling action, by definition, stochastically decreases the average number of packets successfully transmitted in the period. This is a unique "exploitation vs exploration" tradeoff that exists at the period level. This tradeoff exists in addition to the classic one at the slot level [57–59]. The last statement is evidenced by the fact that, in the high traffic region, i.e., when the number of packets in a slot is at least as high as the number of slots in a period (when $N \geq T$), the *last feedback slot* is always the last slot of the period and the tradeoff at the period level vanishes, while the scheduling problem is still a sequential decision process with the classic tradeoff at the slot level.

Fig. 3.2 illustrates the period-level tradeoff discussed above. Consider two arbitrary policies $\mathcal{P}_1$ and $\mathcal{P}_2$. In the current period, policy $\mathcal{P}_1$ idles in the first slot and schedules users in the order $\{1, 2, 3\}$ with immediate retransmission after failure of a packet. Policy $\mathcal{P}_2$ does not idle and schedules users in the same order as $\mathcal{P}_1$, with immediate retransmissions. Let $N = 3$ and $T = 4$. Consider sample path S_1 of the underlying channels. Along S_1, both policies achieve the same number of successful transmissions. Under policy $\mathcal{P}_1$, thanks to idling, the scheduler has more fresh CSI in the next period, compared to the case under policy $\mathcal{P}_2$. Thus idling is beneficial along S_1. On the other hand, along sample path S_2, policy $\mathcal{P}_1$, thanks to idling, achieves lower number of successful transmissions than policy $\mathcal{P}_2$, while both policies provide CSI that is equally fresh to the scheduler, in the next period. Thus idling is disadvantageous along S_2. This example illustrates the period-level "exploitation vs exploration" tradeoff inherent in the scheduling problem at hand and also suggests that the optimal policy is not necessarily non-idling.

3.4 Idling as the Optimal Action: A Case Study

As discussed in the previous section, for certain (underlying, unobservable) sample paths, idling may result in higher system rewards than that of transmitting. In what follows, we show, using numerical example, that under certain system states, idling is indeed an optimal decision.

Consider a two-user system with $K = 2$ and $T = 3$, with the priority distribution given by

$$\mathbf{Q} = \begin{cases} 20, & w.p.\ 0.6 \\ 10, & w.p.\ 0.3 \\ 1, & w.p.\ 0.1 \end{cases}.$$

For this specific system, we focus on the optimal decision at the second slot of the first period, i.e., $t = 2$, $\tau = 1$. Let the current system state be such that $\mathcal{S}_{t=2} = \{1\}$, i.e., the packet of user 2 has already been successfully transmitted, and the current priorities of packets are given by $q_{\tau=1}(1) = 1$ and $q_{\tau=1}(2)$ is arbitrary. Also, let the belief values be such that $\pi_{t=2} = [0.32, 0.6]$[1]. Now, the scheduler can either schedule the packet of user 1 to maximize the expected reward in the current period, or *idle* in the current slot to improve the expected reward in the next period. In Table 3.1, for the considered system state, we record the loss in the current period and the gain in the next period, due to idling (over transmitting) decision in the current slot. As seen in Table 3.1, upon idling in the current slot, the loss in the current period is compensated by the gain in the next period, eventually making idling the optimal decision in the considered setting. This case study illustrates the existence of the two-level tradeoffs in the scheduling problem at hand. Note that, in contrast, in the high traffic region, i.e., when $N \geq T$, as discussed in Section 3.3, the period level tradeoff vanishes, essentially rendering idling trivially suboptimal.

Note that the memory in the Markov channels, broadly defined as $(p - r)$, is critical in deciding the gain in the next period corresponding to the idling decision. Indeed, when the channel has no memory, idling is trivially sub-optimal. In Table 3.2, we compare the effect of system memory on the total reward gain brought by idling against transmitting in the original asymmetric system and a special case of the symmetric system, respectively. The symmetric system is defined as follows: the channel between base station and each user is *i.i.d.* across users, i.e., $p_i = p$, $r_i = r$, $\forall\ i$, and each user's flow generates packets with deterministic and identical priority, i.e., $\mathbf{Q}$ is a

[1]Note that the belief values are chosen to be consistent with the Markov channel statistics.

Table 3.1: Comparison of the loss in the current period and the gain in the next period due to idling (against transmitting) in the current slot. Markov channel parameters used: $p_1 = 0.8$, $r_1 = 0.2$, $p_2 = 0.6$, $r_2 = 0.4$.

Period-level Tradeoff	Idling − Transmitting
Reward Loss in Current Period	−0.0640
Reward Gain in Next Period	0.0898
Total Reward Gain	0.0258

Table 3.2: Illustration of the effect of system memory on the total reward gain due to idling in the asymmetric and symmetric systems.

Memory	Total Reward Gain (Idling − Transmitting)	
$(p_1 - r_1)$	Asymmetric System	Symmetric System
0.1	−0.1611	−0.2045
0.4	0.0091	−0.1037
0.7	0.0240	−0.0396

constant. Without loss of generality, we set $\mathbf{Q} = 1$ in the symmetric system. For ease of illustration, in the asymmetric system, we fix the user 2's channel memory, and study the effect of user 1's channel memory on the gain due to idling. Also, in the asymmetric system, we apply the same set of parameters used in Table 3.1. It is expected that the benefit of "more fresh" feedback brought by idling increases with the memory. This is illustrated in Table 3.2, where, as the memory increases, the total reward gain from idling increases for both the asymmetric and symmetric systems. Also, for the asymmetric system, there exists a threshold, such that, for memory above this threshold, idling is optimal, and otherwise it is not. However, interestingly, for the symmetric system, notwithstanding the presence of the period-level "exploitation vs exploration" tradeoff, the idling decision appears to be suboptimal. Motivated by this last observation, we take a closer look at the symmetric two-user system in the next section.

For the symmetric two-user system, it turns out that, despite the complicated dynamics of the scheduling problem, a non-idling greedy policy is optimal. We now define the greedy policy formally.

Definition 3.5.1. *The greedy policy maximizes the expected immediate reward, without considering the future rewards. Specifically, the greedy decision in slot t is given by*

$$\hat{a}_t = \underset{a_t \in \{0,1,\cdots,N\}}{\arg\max} R_t(\pi_t, \mathcal{S}_t; a_t), \tag{3.6}$$

where $R_t(\cdot)$ is the immediate reward defined in (3.3).[2]

If the system has undelivered packets, the greedy policy schedules the user with the highest channel belief value among those with undelivered packets. Thus the greedy policy is, by definition, non-idling when the system is backlogged.

We now focus on the optimality properties of the greedy policy in the symmetric two-user system. We first show an equivalence between the greedy policy in the original system and a genie-aided system. The genie-aided system is defined as follows.

Definition 3.5.2. *In the genie-aided system, at the end of each slot, the scheduler learns about the current channel state of all the downlink users, if a user was scheduled in that slot. Contrast this with the original system where the scheduler learns, via feedback, only the channel state of the currently scheduled user.*

[2]Since in symmetric systems, the priorities are all assumed to be 1, we omit $q_{\tau(t)}$ in the equations hereafter for ease of exposition.

We now report our result on the equivalence between the original system and the genie-aided system under the greedy policy.

Proposition 3.5.1. *When $N = 2$, under the greedy policy, the total discounted reward in the original system equals that of the genie-aided system.*

Remarks and Proof Outline: The proof follows from the observation that, due to the special "toggle" property of the greedy policy in the two-user system, observing any one of the channels in a fixed past slot renders the same greedy choice (and the associated reward) in the current slot, as when both the channels were observed. The complete proof is given in Appendix 3.7.1.

We now report our results on the system level conditions under which the greedy policy is optimal in the two-user genie-aided system.

Lemma 3.5.1. *A necessary and sufficient condition ($\mathcal{C}$) for the optimality of the greedy policy in the two-user genie-aided system is given as follows: $\forall\ \pi_t \in \{x, y\}$, with $x, y \in \{p, r\}$, and $\forall\ v \neq u$, with $u \in \{\emptyset, 1, 2\}$, $v \in \{1, 2\}$,*

$$1 + \beta W_t(\pi_t, \{u\}) \geq \beta W_t(\pi_t, \{u, v\}),\ \textit{for all } t, \tag{3.7}$$

where $W_t(\pi_t, \mathcal{S}_t)$ denotes the total discounted reward in slot t, in the genie-aided system, with π_t denoting the belief values of the two users and $\mathcal{S}_t$ denoting the set of users with undelivered packets in slot t.

Remarks and Proof Outline: (3.7) essentially captures the period-level tradeoff discussed in Section 3.3. The right hand side (RHS) corresponds, in a broad sense, to the case when the system has one more packet to transmit than the left hand side (LHS), thanks to an idling operation in the past. Thus the left and right hand sides loosely correspond, respectively, to the period-level

"exploitation" and "exploration" sides of the tradeoff discussed in Section 3.3. If $\mathcal{C}$ holds, then the suboptimality of the idling operation can be established using induction arguments. This, along with the fact that full feedback is received at the end of the slot, can be used to establish the optimality of the greedy policy in the two-user genie-aided system. The complete proof is given in Appendix 3.7.2.

Proposition 3.5.2. *The greedy policy is optimal in the symmetric two-user system.*

Proof Outline: The proof proceeds by first showing that the condition $\mathcal{C}$ is satisfied, indicating that the greedy policy is optimal in the two-user genie-aided system. Next, using the equivalence established in Proposition 3.5.1, the greedy policy is optimal in the original two-user system. A key step in the proof is to establish the suboptimality of the idling action when there is only one packet undelivered, i.e., $1 + \beta W_t(\pi_t, \emptyset) \geq \beta W_t(\pi_t, \{v\}), \forall\ v \in \{1, 2\},\ t \in \{1, \cdots, KT\}$. The complete proof is given in Appendix 3.7.2.

3.6 Conclusion

We explored the optimization of joint channel learning and scheduling of real-time traffic with hard deadlines and packet-level priorities, under Markovian channels. Formulating the scheduling problem as a partially observable Markov decision process, we identified the unique two-timescale nature of the "exploitation vs exploration" tradeoffs, where idling provides a new dimension to the action space. Via a case study, we showed that idling is an optimal action under certain system states, further underlining the fundamental tradeoffs between exploitation and exploration/idling. In contrast, for the special case

with a symmetric two-user system, we showed that the scheduling problem exhibits a unique structure – the equivalence with the genie-aided system – that renders the *non-idling* greedy policy optimal.

3.7 Appendix

3.7.1 Proof of Proposition 3.5.1

Under greedy policy, the total discounted reward in the original system is equal to that in the genie-aided system when $N = 2$, i.e.,

$$V_t(\pi_t, \mathcal{S}_t; \hat{\mathcal{G}}) = W_t(\pi_t, \mathcal{S}_t; \hat{\mathcal{G}}),$$

where $\hat{\mathcal{G}}$ denotes the greedy policy.

We prove this by using induction. Consider the t_1th and t_2th slots in the whole horizon, such that $1 \leq t_1 \leq t_2 \leq KT$. Let $\hat{a}_t$ and $\hat{b}_t$ denote the scheduling decision in slot t under the greedy policy in the original system and in the genie-aided system respectively. Note that $\hat{a}_t, \hat{b}_t \in \{0, 1, 2\}$.

For any underlying channel sample path $\{C_{t_1}, \cdots, C_{t_2}\}$ with $C_l = \{C_l(1), C_l(2)\}$, we wish to show that

$$\hat{a}_l = \hat{b}_l, \forall\ l \in \{t_1, \cdots, t_2\}. \tag{3.8}$$

Obviously, it is true for $t_1 = t_2$. For $t_1 < t_2$, we assume that (3.8) holds for any $l \in \{t_1, \cdots, t_2 - 1\}$. Then we only need to show $\hat{a}_{t_2} = \hat{b}_{t_2}$.

Let t, $t_1 \leq t < t_2$, be the latest slot in which a user was scheduled, i.e., $\hat{a}_t \neq 0$ and $\hat{a}_l = 0$, $\forall\ l \in \{t + 1, \cdots, t_2 - 1\}$. Let f_t denote the feedback received from user $\hat{a}_t$ at the end of slot t. Let π_t denote the belief value of the users in slot t when the decision $\hat{a}_t$ was made.

The current belief values π_{t_2} is now given by

$$\pi_{t_2}(\hat{a}_t) = \begin{cases} \mathcal{T}^{t_2-t-1}(p), & \text{if } f_t = 1 \\ \mathcal{T}^{t_2-t-1}(r), & \text{if } f_t = 0 \end{cases}. \tag{3.9}$$

Let $\tilde{a}_t$ denote the user not scheduled in slot t. Then $\pi_{t_2}(\tilde{a}_t) = \mathcal{T}^{t_2-t}(\pi_t(\tilde{a}_t))$. Based on the fact that $\mathcal{T}^m(p) \geq \mathcal{T}^{m+1}(x) \geq \mathcal{T}^m(r)$, $\forall x \in [0,1]$, the scheduling decision in slot t_2 in the original system is given by

$$\hat{a}_{t_2} = \begin{cases} \hat{a}_t, & \text{if } f_t = 1, \mathcal{S}_{t_2} = \{1,2\} \\ \tilde{a}_t, & \text{if } f_t = 0, \mathcal{S}_{t_2} = \{1,2\} \\ 1, & \text{if } \mathcal{S}_{t_2} = \{1\} \\ 2, & \text{if } \mathcal{S}_{t_2} = \{2\} \\ 0, & \text{if } \mathcal{S}_{t_2} = \emptyset \end{cases}. \tag{3.10}$$

Similarly to the original system, we define f'_t and π'_t as the feedback and the belief value for the genie-aided system respectively. Note that in the genie-aided system, both users will feedback to the scheduler, and hence $\pi'_t(i)$ can be either p or r. Then the scheduling decision in slot t_2 in the genie-aided system is given by

$$\hat{b}_{t_2} = \begin{cases} \hat{b}_t, & \text{if } f'_t(\hat{b}_t) = 1, \mathcal{S}'_{t_2} = \{1,2\} \\ \tilde{b}_t, & \text{if } f'_t(\hat{b}_t) = 0, \mathcal{S}'_{t_2} = \{1,2\} \\ 1, & \text{if } \mathcal{S}'_{t_2} = \{1\} \\ 2, & \text{if } \mathcal{S}'_{t_2} = \{2\} \\ 0, & \text{if } \mathcal{S}'_{t_2} = \emptyset \end{cases}. \tag{3.11}$$

where $\hat{b}_t$ is the user scheduled in slot t and $\tilde{b}_t$ is the user not scheduled in slot t. Note that when both users returned ACK feedback in slot t, we choose the user $\hat{b}_t$ in slot t_2 if $\mathcal{S}'_{t_2} = \{1,2\}$ without loss of generality.

Due to the induction that, for the underlying channel sample path, $\hat{a}_l = \hat{b}_l, \forall\ l \in \{t_1, \cdots, t_2 - 1\}$, we have $\mathcal{S}_{t_2} = \mathcal{S}'_{t_2}$ and $\hat{a}_t = \hat{b}_t$. Therefore,

$\hat{a}_{t_2} = \hat{b}_{t_2}$. The preceding proof is based on the assumption that such t exists. If there does not exist such t, i.e., the greedy policy idles in all slots before t_2, the proof holds trivially. Since for any t_1 and t_2 within the horizon, (3.8) holds, therefore the proof is concluded.

3.7.2 Proof of Lemma 3.5.1 and Proposition 3.5.2

To show Proposition 3.5.2, it is equivalent to show that the greedy policy is optimal in the genie-aided system based on Proposition 3.5.1. Mathematically, it means that in any slot t,

$$W_t(\pi_t, \mathcal{S}_t; a_t \in \mathcal{S}_t) \geq W_t(\pi_t, \mathcal{S}_t; a_t = 0), \tag{3.12a}$$

$$W_t(\pi_t, \mathcal{S}_t; a_t = i) \geq W_t(\pi_t, \mathcal{S}_t; a_t = j) \quad \text{for } \forall i, j \in \mathcal{S}_t \text{ and } \pi_t(i) \geq \pi_t(j). \tag{3.12b}$$

In what follows, we will show that (3.7) is the necessary and sufficient condition for (3.12)(i.e., Lemma 3.5.1), and (3.7) holds, i.e., Proposition 3.5.2 holds. The proof is based on double backward induction on t.

For the last time slot $t = KT$, it is easy to verify (3.12) and (3.7) for $\mathcal{S}_t = \{1, 2\}, \{1\}$ and $\{2\}$. Now suppose (3.12) and (3.7) hold for all $t + 1, \cdots, KT$. In what follows, we show (3.12) and (3.7) also hold for t.

The proof for (3.12) at time t:

1) $\mathcal{S}_t = \{1, 2\}$

Without loss of generality, assume $x \geq y$. Let $X(1) = W_t(x, y, \{1, 2\}; a_t = 1)$, $X(2) = W_t(x, y, \{1, 2\}; a_t = 2)$, and $X(0) = W_t(x, y, \{1, 2\}; a_t = 0) = \beta W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1, 2\})$. When $t \neq lT$, where $l = 1, \cdots, K$, i.e., $t + 1$ is not

the beginning of a period, we have

$$\begin{aligned} X(1) &= xy(1+\beta W_{t+1}(p,p,\{2\})) + x(1-y)(1+\beta W_{t+1}(p,r,\{2\})) \\ &\quad +(1-x)y\beta W_{t+1}(r,p,\{1,2\}) + (1-x)(1-y)\beta W_{t+1}(r,r,\{1,2\}) \\ X(2) &= xy(1+\beta W_{t+1}(p,p,\{2\})) + x(1-y)\beta W_{t+1}(p,r,\{2\}) \\ &\quad +(1-x)y(1+\beta W_{t+1}(r,p,\{1,2\})) + (1-x)(1-y)\beta W_{t+1}(r,r,\{1,2\}). \end{aligned} \tag{3.13}$$

Then we have

$$X(1) - X(2) = (x-y)(1+\beta W_{t+1}(p,p,\{2\}) - \beta W_{t+1}(p,p,\{1,2\})) \geq 0, \tag{3.14}$$

where $1+\beta W_{t+1}(p,p,\{2\}) - \beta W_{t+1}(p,p,\{1,2\}) \geq 0$ is due to the induction on (3.7).

Next, we show $X(1) - X(0) \geq 0$.

$$\begin{aligned} & X(1) - X(0) \\ = \; & xy(1+\beta W_{t+1}(p,p,\{2\}) - \beta W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ & +x(1-y)(1+\beta W_{t+1}(p,r,\{2\}) - \beta W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ & +(1-x)y\beta(W_{t+1}(r,p,\{1,2\}) - W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ & +(1-x)(1-y)\beta(W_{t+1}(r,r,\{1,2\}) - W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ \overset{(a)}{\geq} \; & xy\beta(W_{t+1}(p,p,\{1,2\}) - W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ & +x(1-y)\beta(W_{t+1}(p,r,\{1,2\}) - W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ & +(1-x)y\beta(W_{t+1}(r,p,\{1,2\}) - W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ & +(1-x)(1-y)\beta(W_{t+1}(r,r,\{1,2\}) - W_{t+1}(\mathcal{T}(x),\mathcal{T}(y),\{1,2\})) \\ \overset{(b)}{\geq} \; & 0, \end{aligned} \tag{3.15}$$

where (a) is due to the induction on (3.7); (b) by expanding W_{t+1} in terms of W_{t+2}, we can easily verify this. Intuitively speaking, it is easy to see that (b)

holds, due to the fact that the reward with channel feedback is greater than that without feedback.

2) $\mathcal{S}_t = \{1\}$

Let $X(1) = W_t(x, y, \{1\}; a_t = 1)$ and $X(0) = W_t(x, y, \{1\}; a_t = 0) = \beta W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})$.

$$\begin{aligned}
& X(1) - X(0) \\
= \; & xy(1 + \beta W_{t+1}(p, p, \emptyset) - \beta W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
& + x(1-y)(1 + \beta W_{t+1}(p, r, \emptyset) - \beta W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
& + (1-x)y\beta(W_{t+1}(r, p, \{1\}) - W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
& + (1-x)(1-y)\beta(W_{t+1}(r, r, \{1\}) - W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
\overset{(a)}{\geq} \; & xy\beta(W_{t+1}(p, p, \{1\}) - W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
& + x(1-y)\beta(W_{t+1}(p, r, \{1\}) - W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
& + (1-x)y\beta(W_{t+1}(r, p, \{1\}) - W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
& + (1-x)(1-y)\beta(W_{t+1}(r, r, \{1\}) - W_{t+1}(\mathcal{T}(x), \mathcal{T}(y), \{1\})) \\
\overset{(b)}{\geq} \; & 0,
\end{aligned}$$

where the reasons for (a) and (b) are the same as (3.15). Similarly, when $\mathcal{S}_t = \{2\}$, (3.12) also holds.

When $t = lT$, where $l = 1, \cdots, K$, i.e., $t + 1$ is the beginning of a period, it is easy to verify that (3.12) also holds.

Now we can turn our focus to the proof for (3.7) at time t, which relies on the results of (3.12) at time t.

The proof for (3.7) at time t:

First, we check the case of $t \neq lT$, where $l = 1, \cdots, K$.

1) $\mathcal{S}_t = \{1, 2\}$

In this case, we first consider $\mathcal{S}'_t = \{1\}$. Without loss of generality, we assume that $x \geq y$.

$$\begin{aligned}
& 1 + \beta W_t(x, y, \{1\}) - \beta W_t(x, y, \{1, 2\}) \\
\overset{(a)}{=} \; & 1 - \beta + xy\beta(1 + \beta W_{t+1}(p, p, \emptyset) - \beta W_{t+1}(p, p, \{2\})) \\
& + x(1-y)\beta(1 + \beta W_{t+1}(p, r, \emptyset) - \beta W_{t+1}(p, r, \{2\})) \\
& + (1-x)y\beta(1 + \beta W_{t+1}(r, p, \{1\}) - \beta W_{t+1}(r, p, \{1, 2\})) \\
& + (1-x)(1-y)\beta(1 + \beta W_{t+1}(r, r, \{1\}) - \beta W_{t+1}(x, y, \{1, 2\})) \\
\overset{(b)}{\geq} \; & 0,
\end{aligned}$$

where (a) is due to the results of (3.12) at time t; (b) is due to the induction of (3.7). Similarly, it is easy to verify the case of $\mathcal{S}'_t = \{2\}$.

2) $\mathcal{S}_t = \{1\}$

In this case, we only need to consider the case of $\mathcal{S}'_t = \emptyset$. Without loss of generality, we only need to show

$$1 + \beta W_t(x, y, \emptyset) \geq \beta W_t(x, y, \{1\}). \tag{3.16}$$

In what follows, we prove that (3.16) holds.

When the slot t is in the last period, the inequality holds trivially. In what follows, we consider the case that the slot t is not in the last period. Let the slot $(t + L)$ be the beginning slot of the next period, where $L \geq 1$. When $L = 0$, the inequality holds trivially. For $L = 1$, we can also show that the inequality holds by expanding each term in term of the reward in the slot $(t + 2)$. Thus, in what follows, we consider the case $L \geq 2$.

Let $LHS = 1 + W_t(x, y, \emptyset)$ and $RHS = W_t(x, y, \{1\})$. Then we can easily write LHS in terms of the reward in the slot $(t + L)$ as follows:

$$LHS = 1 + W_{t+L}(\mathcal{T}^L(x), \mathcal{T}^L(y), \{1, 2\}).$$

For the RHS, we can also write it in terms of the reward in the slot $(t+L)$ according to all possible sample paths, and sort them out based on whether the packet of user 1 is delivered before the slot $(t+L)$,

$$\begin{aligned}
RHS = & x\Big(y(1+W_{t+L}(\mathcal{T}^{L-1}(p),\mathcal{T}^{L-1}(p),\{1,2\})) \\
& \quad + (1-y)(1+W_{t+L}(\mathcal{T}^{L-1}(p),\mathcal{T}^{L-1}(r),\{1,2\}))\Big) \\
& + \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r\Big(\mathcal{T}^{k+1}(y)(1+W_{t+L}(\mathcal{T}^{L-k-2}(p),\mathcal{T}^{L-k-2}(p),\{1,2\})) \\
& \quad + (1-\mathcal{T}^{k+1}(y))(1+W_{t+L}(\mathcal{T}^{L-k-2}(p),\mathcal{T}^{L-k-2}(r),\{1,2\}))\Big) \\
& + (1-x)(1-r)^{L-1}\Big(\mathcal{T}^{L-1}(y)W_{t+L}(r,p,\{1,2\}) \\
& \quad + (1-\mathcal{T}^{L-1}(y))W_{t+L}(r,r,\{1,2\})\Big),
\end{aligned} \tag{3.17}$$

where the first and second terms denote the rewards that the packet of user 1 is delivered before the slot $(t+L)$, whereas the third term denotes the reward that the packet of user 1 is not delivered before the slot $(t+L)$. Further, (3.17) can be also written as

$$\begin{aligned}
RHS \;=\; & x\Big(1+W_{t+L}(\mathcal{T}^{L-1}(p),\mathcal{T}^{L}(y),\{1,2\})\Big) \\
& + \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r\Big(1+W_{t+L}(\mathcal{T}^{L-k-2}(p),\mathcal{T}^{L}(y),\{1,2\})\Big) \\
& + (1-x)(1-r)^{L-1}W_{t+L}(r,\mathcal{T}^{L}(y),\{1,2\}),
\end{aligned} \tag{3.18}$$

due to the fact that, for each term, we can combine those sample paths that lead to the same scheduling decision in slot $(t+L+1)$.

Next, we proceed to show $LHS - RHS \geq 0$ by expanding LHS and RHS in terms of the future rewards in slot $(t+L+1)$.

First, we consider the case of $x \geq y$. In this case, $LHS - RHS$ can be written as the sum of immediate reward and the future reward in slot

$(t+L+1)$.

$$
\begin{aligned}
& LHS - RHS \\
= \; & 1 + \mathcal{T}^L(x) + \mathcal{T}^L(x)\mathcal{T}^L(y) W_{t+L+1}(p,p,\{2\}) \\
& + \mathcal{T}^L(x)(1-\mathcal{T}^L(y)) W_{t+L+1}(p,r,\{2\}) \\
& + (1-\mathcal{T}^L(x))\mathcal{T}^L(y) W_{t+L+1}(r,p,\{1,2\}) \\
& + (1-\mathcal{T}^L(x))(1-\mathcal{T}^L(y)) W_{t+L+1}(r,r,\{1,2\}) \qquad (3.19) \\
& - x\Big(1 + \mathcal{T}^{L-1}(p) + \mathcal{T}^{L-1}(p)\mathcal{T}^L(y) W_{t+L+1}(p,p,\{2\}) \\
& + \mathcal{T}^{L-1}(p)(1-\mathcal{T}^L(y)) W_{t+L+1}(p,r,\{2\}) \\
& + (1-\mathcal{T}^{L-1}(p))\mathcal{T}^L(y) W_{t+L+1}(r,p,\{1,2\}) \\
& + (1-\mathcal{T}^{L-1}(p))(1-\mathcal{T}^L(y)) W_{t+L+1}(r,r,\{1,2\})\Big) \\
& - \textstyle\sum_{k=0}^{L-2} (1-x)(1-r)^k r\Big(1 + \mathcal{T}^{L-k-2}(p) + \mathcal{T}^{L-k-2}(p)\mathcal{T}^L(y) W_{t+L+1}(p,p,\{2\}) \\
& + \mathcal{T}^{L-k-2}(p)(1-\mathcal{T}^L(y)) W_{t+L+1}(p,r,\{2\}) \\
& + (1-\mathcal{T}^{L-k-2}(p))\mathcal{T}^L(y) W_{t+L+1}(r,p,\{1,2\}) \\
& + (1-\mathcal{T}^{L-k-2}(p))(1-\mathcal{T}^L(y)) W_{t+L+1}(r,r,\{1,2\})\Big) \\
& - (1-x)(1-r)^{L-1}\Big(\mathcal{T}^L(y) + r\mathcal{T}^L(y) W_{t+L+1}(p,p,\{1\}) \\
& + r(1-\mathcal{T}^L(y)) W_{t+L+1}(p,r,\{1,2\}) \\
& + (1-r)\mathcal{T}^L(y) W_{t+L+1}(r,p,\{1\}) \\
& + (1-r)(1-\mathcal{T}^L(y)) W_{t+L+1}(r,r,\{1,2\})\Big).
\end{aligned}
$$

Since (3.19) is complicated, in what follows, we compute the immediate reward, i.e., the constant in (3.19), and the coefficients of $W_{t+L+1}(p,p,\{2\})(= W_{t+L+1}(p,p,\{1\}))$, $W_{t+L+1}(p,r,\{2\})(= W_{t+L+1}(r,p,\{1\}))$, $W_{t+L+1}(r,p,\{1,2\})(=$

$W_{t+L+1}(p, r, \{1, 2\})$), and $W_{t+L+1}(r, r, \{1, 2\})$ separately.

$$
\begin{aligned}
& \text{constant in (3.19)} \\
= \;& 1 + \mathcal{T}^L(x) - x(1 + \mathcal{T}^{L-1}(p)) - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r(1 + \mathcal{T}^{L-k-2}(p)) \\
& -(1-x)(1-r)^{L-1}\mathcal{T}^L(y) \\
= \;& 1 - x + \mathcal{T}^L(x) - x\mathcal{T}^{L-1}(p) - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r(1 + \mathcal{T}^{L-k-2}(p)) \\
& -(1-x)(1-r)^{L-1}\mathcal{T}^L(y) \\
\overset{(a)}{=} \;& (1-x) + (1-x)\mathcal{T}^{L-1}(r) - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r(1 + \mathcal{T}^{L-k-2}(p)) \\
& -(1-x)(1-r)^{L-1}\mathcal{T}^L(y) \\
= \;& (1-x)\Big(1 + \mathcal{T}^{L-1}(r) - \textstyle\sum_{k=0}^{L-2}(1-r)^k r(1 + \mathcal{T}^{L-k-2}(p)) - (1-r)^{L-1}\mathcal{T}^L(y)\Big) \\
\overset{(b)}{=} \;& (1-x)\Big(1 - \textstyle\sum_{k=0}^{L-2}(1-r)^k r - (1-r)^{L-1}\mathcal{T}^L(y) + \mathcal{T}^{L-1}(r) \\
& - \textstyle\sum_{k=0}^{L-2}(1-r)^k r\mathcal{T}^{L-k-2}(p)\Big) \\
= \;& (1-x)\Big((1-r)^{L-1}(1 - \mathcal{T}^L(y)) + (1-r)^{L-1} r\Big) \\
= \;& (1-x)(1-r)^{L-1}(1 + r - \mathcal{T}^L(y)),
\end{aligned}
\tag{3.20}
$$

where (a) and (b) are based on the fact $\mathcal{T}^k(x) = x\mathcal{T}^{k-1}(p) + (1-x)\mathcal{T}^{k-1}(r)$, which will be frequently used in the following derivation.

$$
\begin{aligned}
& \text{coefficient of } W_{t+L+1}(p, p, \{2\}) \text{ in (3.19)} \\
= \;& \mathcal{T}^L(x)\mathcal{T}^L(y) - x\mathcal{T}^{L-1}(p)\mathcal{T}^L(y) - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r\mathcal{T}^{L-k-2}(p)\mathcal{T}^L(y) \\
& -(1-x)(1-r)^{L-1} r\mathcal{T}^L(y) \\
= \;& \mathcal{T}^L(y)\Big(\mathcal{T}^L(x) - x\mathcal{T}^{L-1}(p) - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r\mathcal{T}^{L-k-2}(p) \\
& -(1-x)(1-r)^{L-1} r\Big) \\
= \;& 0
\end{aligned}
\tag{3.21}
$$

$$
\begin{aligned}
& \text{coefficient of } W_{t+L+1}(p, r, \{2\}) \text{ in (3.19)} \\
= \;& \mathcal{T}^L(x)(1-\mathcal{T}^L(y)) - x\mathcal{T}^{L-1}(p)(1-\mathcal{T}^L(y)) \\
& - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r\mathcal{T}^{L-k-2}(p)(1-\mathcal{T}^L(y)) \\
& -(1-x)(1-r)^{L-1}(1-r)\mathcal{T}^L(y) \\
= \;& (1-x)(1-r)^{L-1}r(1-\mathcal{T}^L(y)) - (1-x)(1-r)^{L-1}(1-r)\mathcal{T}^L(y) \\
= \;& (1-x)(1-r)^{L-1}(r-\mathcal{T}^L(y))
\end{aligned}
\tag{3.22}
$$

$$
\begin{aligned}
& \text{coefficient of } W_{t+L+1}(r, p, \{1,2\}) \text{ in (3.19)} \\
= \;& (1-\mathcal{T}^L(x))\mathcal{T}^L(y) - x(1-\mathcal{T}^{L-1}(p))\mathcal{T}^L(y) \\
& - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r(1-\mathcal{T}^{L-k-2}(p))\mathcal{T}^L(y) \\
& -(1-x)(1-r)^{L-1}r(1-\mathcal{T}^L(y)) \\
= \;& (1-x)(1-r)^{L-1}(\mathcal{T}^L(y)-r)
\end{aligned}
\tag{3.23}
$$

$$
\begin{aligned}
& \text{coefficient of } W_{t+L+1}(r, r, \{1,2\}) \text{ in (3.19)} \\
= \;& (1-\mathcal{T}^L(x))(1-\mathcal{T}^L(y)) - x(1-\mathcal{T}^{L-1}(p))(1-\mathcal{T}^L(y)) \\
& - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r(1-\mathcal{T}^{L-k-2}(p))(1-\mathcal{T}^L(y)) \\
& -(1-x)(1-r)^{L-1}(1-r)(1-\mathcal{T}^L(y)) \\
= \;& 0
\end{aligned}
\tag{3.24}
$$

Therefore, when $x \geq y$, $LHS - RHS$ can be written as

$$
\begin{aligned}
& LHS - RHS \\
= \;& (1-x)(1-r)^{L-1}(1+r-\mathcal{T}^L(y)) \\
& +(1-x)(1-r)^{L-1}(\mathcal{T}^L(y)-r)(W_{t+L+1}(p, r, \{1,2\}) - W_{t+L+1}(p, r, \{2\})) \\
\overset{(a)}{\geq} \;& 0,
\end{aligned}
\tag{3.25}
$$

where (a) is due to the facts that $(1-x)(1-r)^{L-1}(1+r-\mathcal{T}^L(y)) \geq 0$, $\mathcal{T}^L(y) \geq r$, and $W_{t+L+1}(p, r, \{1,2\}) \geq W_{t+L+1}(p, r, \{2\})$.

When $x < y$, we use the same approach as in the case of $x \geq y$ to compute $LHS - RHS$. In this case, RHS is the same as in the case of $x \geq y$,

while LHS is equal to the reward of scheduling user 2 in slot (t+L). Therefore, $LHS - RHS$ can be written as

$$
\begin{aligned}
& LHS - RHS \\
= \; & 1 + \mathcal{T}^L(y) + \mathcal{T}^L(x)\mathcal{T}^L(y)W_{t+L+1}(p,p,\{1\}) \\
& + \mathcal{T}^L(x)(1-\mathcal{T}^L(y))W_{t+L+1}(p,r,\{1,2\}) \\
& + (1-\mathcal{T}^L(x))\mathcal{T}^L(y)W_{t+L+1}(r,p,\{1\}) \\
& + (1-\mathcal{T}^L(x))(1-\mathcal{T}^L(y))W_{t+L+1}(r,r,\{1,2\}) \\
& - x\Big(1 + \mathcal{T}^{L-1}(p) + \mathcal{T}^{L-1}(p)\mathcal{T}^L(y)W_{t+L+1}(p,p,\{2\}) \\
& + \mathcal{T}^{L-1}(p)(1-\mathcal{T}^L(y))W_{t+L+1}(p,r,\{2\}) \\
& + (1-\mathcal{T}^{L-1}(p))\mathcal{T}^L(y)W_{t+L+1}(r,p,\{1,2\}) \\
& + (1-\mathcal{T}^{L-1}(p))(1-\mathcal{T}^L(y))W_{t+L+1}(r,r,\{1,2\})\Big) \\
& - \textstyle\sum_{k=0}^{L-2}(1-x)(1-r)^k r\Big(1 + \mathcal{T}^{L-k-2}(p) \\
& + \mathcal{T}^{L-k-2}(p)\mathcal{T}^L(y)W_{t+L+1}(p,p,\{2\}) \\
& + \mathcal{T}^{L-k-2}(p)(1-\mathcal{T}^L(y))W_{t+L+1}(p,r,\{2\}) \\
& + (1-\mathcal{T}^{L-k-2}(p))\mathcal{T}^L(y)W_{t+L+1}(r,p,\{1,2\}) \\
& + (1-\mathcal{T}^{L-k-2}(p))(1-\mathcal{T}^L(y))W_{t+L+1}(r,r,\{1,2\})\Big) \\
& - (1-x)(1-r)^{L-1}\Big(\mathcal{T}^L(y) + r\mathcal{T}^L(y)W_{t+L+1}(p,p,\{1\}) \\
& + r(1-\mathcal{T}^L(y))W_{t+L+1}(p,r,\{1,2\}) \\
& + (1-r)\mathcal{T}^L(y)W_{t+L+1}(r,p,\{1\}) \\
& + (1-r)(1-\mathcal{T}^L(y))W_{t+L+1}(r,r,\{1,2\})\Big).
\end{aligned}
$$

By following similar algebra to the case of $x \geq y$, we have

$$
\begin{aligned}
& LHS - RHS \\
= \; & \mathcal{T}^L(y) - \mathcal{T}^L(x) + (1-x)(1-r)^{L-1}(1+r-\mathcal{T}^L(y)) \\
& + \Big(\mathcal{T}^L(y) - \mathcal{T}^L(x) + (1-x)(1-r)^{L-1}(r-\mathcal{T}^L(y))\Big) W_{t+L+1}(p,r,\{2\}) \\
& - \Big(\mathcal{T}^L(y) - \mathcal{T}^L(x) + (1-x)(1-r)^{L-1}(r-\mathcal{T}^L(y))\Big) W_{t+L+1}(p,r,\{1,2\}) \\
= \; & (1-x)(1-r)^{L-1}(1+r-\mathcal{T}^L(y)) \\
& + (\mathcal{T}^L(y) - \mathcal{T}^L(x))(1 + W_{t+L+1}(p,r,\{2\}) - W_{t+L+1}(p,r,\{1,2\})) \\
& + (1-x)(1-r)^{L-1}(\mathcal{T}^L(y) - r)(W_{t+L+1}(p,r,\{1,2\}) - W_{t+L+1}(p,r,\{2\})) \\
\overset{(a)}{\geq} \; & 0,
\end{aligned}
$$

where (a) is due to the facts that $(1-x)(1-r)^{L-1}(1+r-\mathcal{T}^L(y))$, $\mathcal{T}^L(y) \geq \mathcal{T}^L(x)$, $1 + W_{t+L+1}(p,r,\{2\}) \geq W_{t+L+1}(p,r,\{1,2\})$ (by induction), and $W_{t+L+1}(p,r,\{1,2\}) \geq W_{t+L+1}(p,r,\{2\})$.

Chapter 4

RISK-AWARE DAY-AHEAD SCHEDULING AND REAL-TIME DISPATCH FOR PLUG-IN ELECTRIC VEHICLES

4.1 Introduction

4.1.1 Motivation

Today's transportation sector accounts for a significant portion of petroleum consumption and carbon emissions worldwide. The growing concerns over energy security, the dependence on oil/petroleum and environmental issues are driving the electrification of transportation and the development of plug-in electric vehicle (EV) technology. By using electricity rather than petroleum, electric vehicles and plug-in hybrid electric vehicles reduce the petroleum consumption and the greenhouse gas emissions. The large-scale implementation of EVs is being planned in the near future. The US administration has planned to put one million EVs on the road by 2015.

It has been widely recognized that the high penetration level of EVs would have considerable impact on the existing power system. The analysis of [63–73] has predicted that a significant amount of EV charging will take place during the peak load period, causing possible branch congestions and voltage problems. Studies in [74, 75] have shown that the existing distribution system infrastructure may only support a low EV penetration level. Without proper coordination, the coincidence between peaks of EV and non-EV load will require investment in generation, transmission, and distribution, in order to maintain the reliability of the power system.

Recent work [76] has proposed an operating framework of load schedul-

ing and dispatch for aggregators of EVs to minimize the charging cost of EVs by using the forecasted EV load. Due to the "random" driving activities of EVs, the actual EV load may be different from the forecasted EV load. Therefore, it is difficult to guarantee that the actual load and the scheduled load remain balanced at each instant. And this mismatch between the supply (the scheduled load) and the demand (the actual load) could cause a deviation of zonal frequency from nominal value [77]. Often times, the energy market consists of a day-ahead market and a real-time market for electricity. Simply put, the day-ahead market produces financially binding schedules for the energy generation and consumption one day before the operating day; and the real-time market is used to achieve the balance between the energy amount scheduled day-ahead and the real-time load. Due to the uncertainty of EVs' driving activities, this load mismatch between scheduling and dispatch would cause additional cost beyond the charging cost of EVs.

Further, different from the real-time dispatch for the traditional energy user, whose energy demand can be satisfied instantaneously, the EV's charging has to be carried out over a much longer duration, due to the physical charging rate constraint. Nevertheless, the required charging time of each EV is usually less than the plug-in period, which renders a challenge task for aggregators to minimize the EVs' charging cost and the load mismatch risk in real time.

4.1.2 Summary of Main Contributions

A primary objective of this chapter is to develop an operating framework of load scheduling and dispatch for aggregators of EVs while taking into account the risk (cost) of the load mismatch in both scheduling and dispatch. Our main contributions are summarized as follows:

- We propose a risk-aware day-ahead scheduling framework that minimizes the charging cost and the load mismatch risk, based on the day-ahead prices and the statistic information of EVs' driving activities. Due to the inclusion of the load mismatch risk, the risk-aware day-ahead scheduling problem is nonconvex. By utilizing the *hidden convexity* structure, we discover that it can be recast as a two-stage stochastic linear program, which can be solved by using the L-shaped method [78].

- We propose a distributed risk-aware real-time dispatch algorithm that minimizes the charging cost and the load mismatch risk in real time. By leveraging the dual decomposition, instead of optimizing the charging strategies for each EV, it suffices for the aggregator to compute the shadow prices for each slot. Based on the shadow prices, each EV can optimize its charging strategy in a distributed manner.

- Based on real data, we compare the performance of our risk-aware day-ahead scheduling algorithm with its counterpart without considering the risk. The results corroborate that by considering the risk, our risk-aware day-ahead scheduling algorithm can reduce not only the overall cost for the case without considering the risk, but also the peak demand of EV charging.

4.1.3 Related Work

As the number of EVs is growing, the demand management of EVs in the power system will be a challenging task. The smart EV charging is an emerging research area to resolve the impacts of EV charging on the power system such as overloading, reduced efficiency, power quality, and voltage regulation particularly at the distribution level. Coordinated charging of EVs is a possible

solution to these problems. [79] proposed a coordinated charging algorithm to minimize the power losses, while [80] explored the relationship between feeder losses, load factor, and load variance in the context of coordinated EV charging, and compared the benefits of using these metrics as objective function in minimizing the impacts of EV charging on the connected distribution system. These works addressed the EV charging in a centralized manner. [76] proposed an operating framework of load scheduling and dispatch for aggregators of EVs to minimize the charging cost of EVs by assuming sufficient accuracy on day-ahead EV load. Note that the forecast-based algorithms are vulnerable to the prediction errors. [81] proposed a decentralized EV charging algorithm to minimize the system load variance, which does not suffer from the prediction errors. However, the energy requirement of each EV may not be guaranteed. [82,83] also proposed real-time decentralized charging algorithms, where it is very challenging to obtain performance guarantees.

Compared with the existing works, we considers the prediction errors and guarantee the energy requirement of each EV in the charging problem. Besides, the proposed real-time dispatch algorithm can be implemented in a distributed manner.

The rest of the chapter is organized as follows. In Section 4.2, we introduce the system model and the risk of the load mismatch. In Section 4.3, we propose the risk-aware day-ahead scheduling framework, and develop the distributed risk-aware real-time dispatch algorithm. In Section 4.4, we evaluate the performance of our risk-aware scheduling and dispatch algorithms. The chapter is concluded in Section 4.5.

4.2 System Model

We consider EV charging controlled by aggregators in a residential area during the off-peak period from 10 p.m. to 7 a.m., where the charging period is divided into T slots of equal duration. The length of each time slot corresponds to the operation time scale of the aggregator, e.g., 15 minutes [79].[1]

4.2.1 Supply Model

Consider a system that a number of EVs are served by a single load-serving entity (or system operator). The system operator participates in the day-ahead market to purchase electricity from generators, based on the statistics of EVs collected from the aggregator. As is standard, the system operator decides the amount of power P_t^{da} to schedule on the day-ahead market for each slot t in the following day according to the day-ahead price c_t^{da} and the statistics of EVs' charging demands for slot t. Conditioned on the day-ahead schedule, the system operator further balances the supply and the demand in real-time market by optimizing EVs' charging strategies. We assume that the penetration level of EVs cannot change the day-ahead prices, i.e., c_t^{da} for slot t is given.

4.2.2 Electric Vehicle Model

Consider N EVs in the system, each of which has different energy demand u_i, charging rate $p_i^{\max}$, plug-in time a_i, and charging deadline d_i. Due to the random driving activities of EVs during the daytime, we assume that the energy demand $u_i \in \mathcal{U}_i$ is a random variable, where $\mathcal{U}_i$ is the set of possible

[1]For simplicity of notation, all parameters used in the chapter are scaled according to the length of a slot.

energy demand. Further, we assume that $u_i \leq (d_i - a_i)p_i^{\max}$; otherwise, no charging scheduling can complete the demand of EV i.

4.2.3 Risks in Day-ahead Scheduling

To reduce the charging cost, the aggregator would charge the EVs when the price is low. Let $u = (u_1, \cdots, u_N)$ denote a vector of random energy demand and $\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_N$ denote the corresponding set of possible energy demand of EVs. Given the day-ahead prices c_t^{da} and the energy demand of EVs $u \in \mathcal{U}$, the aggregator minimizes the charging cost of each EV i, $\sum_{t=1}^{T} c_t^{da} p_i^t(u)$, where $p_i^t(u)$ denotes the charging strategy for EV i under the constraints of $\sum_{t=a_i}^{d_i} p_i^t(u) = u_i$, $0 \leq p_i^t(u) \leq p_i^{\max}$, and $p_i^t(u) = 0$ for $t < a_i$ and $t > d_i$.

Due to the random driving activities of EVs, the power scheduled in the day-ahead market may not match the realized demand. Let $P_t = \sum_{i=1}^{N} p_i^t(u)$ denote the realized demand in slot t, which is a random variable depending on the charging strategy and the driving activities of EVs. Obviously, there exist risks in the day-ahead scheduling, due to the unbalance between P_t and P_t^{da} [84]. When the day-ahead schedule P_t^{da} is insufficient to meet the realized demand P_t, the deficit power must purchase from fast start-up generators at a cost c_r, which is assumed to be greater than the day-ahead price, i.e., $c_r \geq c_t^{da}$, for any t. When the day-ahead schedule P_t^{da} is higher than the realized demand P_t, the system operator needs to cancel the over-scheduled power from the base-line generators incurring a penalty c_p per unit for the canceled amount of generation. Then the expected cost in slot t can be expressed as follows.

$$Cost(t) = E_u\Big\{ \underbrace{(c_t^{da} P_t^{da} + c_r(P_t - P_t^{da}))\mathbf{1}_A}_{\text{Real-time demand}\geq\text{Scheduled power}} + \underbrace{(c_t^{da} P_t + c_p(P_t^{da} - P_t))(1 - \mathbf{1}_A)}_{\text{Real-time demand}<\text{Scheduled power}} \Big\}, \tag{4.1}$$

where the indicator function $\mathbf{1}_A$ corresponds to the aforementioned events, i.e.,

$\mathbf{1}_A = 1$ if $P_t \geq P_t^{da}$, otherwise $\mathbf{1}_A = 0$.

4.3 Joint Risk and Charging Cost Optimization

4.3.1 Risk-aware Day-ahead Scheduling

Given the day-ahead prices c_t^{da}, the aggregator minimizes the scheduling risk and the charging cost of EVs. The day-ahead scheduling can be determined by the following optimization program:

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^{T} Cost(t) \\ \text{subject to} & \sum_{t=a_i}^{d_i} p_i^t(u) = u_i, \ \forall \ i, u \\ & 0 \leq p_i^t(u) \leq p_i^{\max}, \ \forall \ i, t, u \\ & p_i^t(u) = 0 \text{ for } t < a_i \text{ and } t > d_i, \ \forall \ i, t, u \\ \text{variables} & \{p_i^t(u), P_t^{da}\}. \end{array} \tag{4.2}$$

The problem (4.2) jointly optimizes the scheduling risk and the charging cost by considering all possible energy demands of EVs. Due to the nonconvexity of $Cost(t)$, (4.2) is nonconvex, which is difficult to solve. By utilizing the *hidden convexity* of (4.2), we can transform (4.2) into a convex form as follows.

Proposition 4.3.1. *The risk-aware day-ahead problem (4.2) can be recast as the following stochastic linear program:*

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^{T} c_p P_t^{da} + E_u\{\sum_{t,i}(c_t^{da} - c_p - c_r)p_i^t(u)\} \\ \text{subject to} & \sum_{t=a_i}^{d_i} p_i^t(u) \leq u_i, \ \forall \ i, u \\ & \sum_i p_i^t(u) \leq P_t^{da}, \ \forall \ t, u \\ & 0 \leq p_i^t(u) \leq p_i^{\max}, \ \forall \ i, t, u \\ & p_i^t(u) = 0 \text{ for } t < a_i \text{ and } t > d_i, \ \forall \ i, t, u \\ \text{variables} & \{p_i^t(u), P_t^{da}\}. \end{array} \tag{4.3}$$

Proof. The proof is given in Appendix 4.6.1. □

Note that the convex form of the risk-aware day-ahead problem (4.3) is essentially a two-stage stochastic linear program with recourse [85]. If the number of realizations of u is small, we can solve the stochastic linear program (4.3) by a deterministic equivalent linear program transformed by expanding the expectation terms in (4.3). However, the number of realizations of u can be large, which can result in a deterministic program with significantly large size. In this case, we apply the L-shaped method [78] to solve (4.3). To this end, we first write (4.3) in the form of a two-stage program as follows.

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^{T} c_p P_t^{da} + E_u\{h(P^{da}, u)\} \\ \text{subject to} & P_t^{da} \geq 0, \ \forall\ t \\ \text{variables} & \{P_t^{da}\}, \end{array} \tag{4.4}$$

where $P^{da} = (P_1^{da}, \cdots, P_T^{da})$ denotes the vector of day-ahead schedule, and $h(P^{da}, u)$ is defined as the value function of the second stage linear program:

$$\begin{array}{ll} \text{minimize} & \sum_{t,i}(c_t^{da} - c_p - c_r)p_i^t(u) \\ \text{subject to} & \sum_{t=a_i}^{d_i} p_i^t(u) \leq u_i, \ \forall\ i \\ & \sum_i p_i^t(u) \leq P_t^{da}, \ \forall\ t \\ & 0 \leq p_i^t(u) \leq p_i^{\max}, \ \forall\ i, t \\ & p_i^t(u) = 0 \text{ for } t < a_i \text{ and } t > d_i, \ \forall\ i, t \\ \text{variables} & \{p_i^t(u)\}. \end{array} \tag{4.5}$$

The idea of the L-shaped method is to approximate $E_u\{h(P^{da}, u)\}$ by piecewise linear functions $\theta(P^{da})$, where

$$\theta(P^{da}) = \max_{j=1,\cdots,J}\{\alpha_j + \beta_j P^{da}\}. \tag{4.6}$$

Detailed descriptions on how to solve (4.4) are presented in Algorithm 3.

Algorithm 3 Risk-aware day-ahead scheduling

Initialization: Set $J = 0$ and $k = 0$.

1) Set $k = k + 1$. Solve the following linear program:

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^{T} c_p P_t^{da} + \theta \\ \text{subject to} & P_t^{da} \geq 0, \ \forall \ t \\ & \theta \geq \alpha_j + \beta_j P^{da}, j = 1, \cdots, J \\ \text{variables} & \{P_t^{da}\}. \end{array} \tag{4.7}$$

Let $(P^{da}(k), \theta(k))$ be the optimal solution to (4.7). If $J = 0$, $\theta(k)$ is set equal to $-\infty$ and is not considered in (4.7).

2) For each $u \in \mathcal{U}$, solve (4.5) by its dual problem, which is defined as follows:

$$\begin{array}{ll} \text{maximize} & -\sum_t w_t^{P^{da}} P_t^{da}(k) - \sum_i w_i^u u_i + \sum_{t,i} w_{t,i}^p p_i^{max} \\ \text{subject to} & c_t^{da} - c_p - c_r + w_t^{P^{da}} + w_i^u - w_{t,i}^p \geq 0, \forall \ t, i \\ & w_t^{P^{da}} \geq 0, w_i^u \geq 0, w_{t,i}^p \geq 0, \forall \ t, i \\ \text{variables} & \{w_t^{P^{da}}, w_i^u, w_{t,i}^p\}, \end{array} \tag{4.8}$$

where $w_t^{P^{da}}$, w_i^u, and $w_{t,i}^p$ are Lagrangian multipliers associated with the constraints $\sum_i p_i^t(u) \leq P_t^{da}(k)$, $\sum_{t=a_i}^{d_i} p_i^t(u) \leq u_i$, and $p_i^t(u) \leq p_i^{\max}$, respectively.

3) Let $(w_t^{P^{da}}(k,u), w_i^u(k,u), w_{t,i}^p(k,u))$ be the optimal solution to (4.8) in the kth iteration for the realization u. Define $\hat{\theta}(k) = E_u\{-\sum_t w_t^{P^{da}}(k,u) P_t^{da}(k) - \sum_i w_i^u(k,u) u_i + \sum_{t,i} w_{t,i}^p(k,u) p_i^{max}\}$. If $\theta(k) \geq \hat{\theta}(k)$, stop; $P^{da}(k)$ is an optimal solution. Otherwise, set $J = J + 1$, and add an additional constraint $\theta \geq \alpha_J + \beta_J P^{da}$ to (4.7), where $\alpha_J = E_u\{-\sum_i w_i^u(k,u) u_i + \sum_{t,i} w_{t,i}^p(k,u) p_i^{max}\}$ and $\beta_J P^{da} = E_u\{-\sum_t w_t^{P^{da}}(k,u) P_t^{da}\}$.

4.3.2 Risk-aware Real-time Dispatch

Due to the uncertainty in the realization, when dispatching the scheduled power, the aggregator needs to take account of the risks and the charging cost. The purpose is to distribute the scheduled power to the EVs with as little deviation from the schedule P_t^{da} as possible. From the National Household Travel Survey (NHTS) dataset [86], the majority of vehicles arrive at home before 10 p.m. Therefore, we assume that when dispatching the power, the

aggregator has the knowledge of each EV's energy demand u_i.

With the information of EVs' energy demand in the operation day, the real-time dispatch problem is equivalent to the second stage program as defined in (4.5). Different from (4.5), we introduce additional variables $\hat{u}_i$ to denote the portion of EV i's demand that can be realized by the day-ahead schedule. Therefore, the term $c_r(u_i - \hat{u}_i)$ can be interpreted as the under-scheduling cost, i.e., the cost of purchasing energy from real-time market to meet the demand of EV i. Therefore, the risk-aware real-time dispatch problem can be written as

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^{T}(c_t^{da} - c_p)\sum_i p_i^t - \sum_i c_r \hat{u}_i \\ \text{subject to} & \sum_{t=a_i}^{d_i} p_i^t \geq \hat{u}_i,\ \forall\ i \\ & 0 \leq \hat{u}_i \leq u_i,\ \forall\ i \\ & \sum_i p_i^t \leq P_t^{da},\ \forall\ t \\ & 0 \leq p_i^t \leq p_i^{\max},\ \forall\ i,t \\ & p_i^t = 0 \text{ for } t < a_i \text{ and } t > d_i,\ \forall\ i,t \\ \text{variables} & \{p_i^t, \hat{u}_i\}. \end{array} \tag{4.9}$$

Since (4.9) is a linear problem, we can solve it by its dual problem. The corresponding Lagrangian function of (4.9) is given by

$$\begin{array}{ll} & L(\lambda, \nu, p, \hat{u}) \\ = & \sum_t (c_t^{da} - c_p)\sum_i p_i^t - \sum_i c_r \hat{u}_i \\ & + \sum_i \lambda_i(\hat{u}_i - \sum_{t=a_i}^{d_i} p_i^t) + \sum_t \nu_t(\sum_i p_i^t - P_t^{da}) \\ = & \sum_{t,i}(c_t^{da} - c_p - \lambda_i + \nu_t)p_i^t - \sum_i (c_r - \lambda_i)\hat{u}_i - \sum_t \nu_t P_t^{da}, \end{array} \tag{4.10}$$

where $\lambda_i \geq 0$ and $\nu_t \geq 0$ are Lagrangian multipliers. The dual function $D(\lambda, \nu)$

is defined as the minimum of the Lagrangian (4.10) over the feasible set,

$$\begin{aligned} \text{minimize} \quad & L(\lambda, \nu, p, \hat{u}) \\ \text{subject to} \quad & 0 \leq \hat{u}_i \leq u_i, \ \forall \ i \\ & 0 \leq p_i^t \leq p_i^{\max}, \ \forall \ i, t \\ & p_i^t = 0 \text{ for } t < a_i \text{ and } t > d_i, \ \forall \ i, t \\ \text{variables} \quad & \{p_i^t, \hat{u}_i\}. \end{aligned} \tag{4.11}$$

The corresponding dual problem is defined as the maximization of the dual function over all nonnegative dual variables,

$$\begin{aligned} \text{maximize} \quad & D(\lambda, \nu) \\ \text{subject to} \quad & \lambda_i \geq 0, \ \forall \ i \\ & \nu_t \geq 0, \ \forall \ t \\ \text{variables} \quad & \{\lambda_i, \nu_t\}. \end{aligned} \tag{4.12}$$

Let k denote the iteration index and $\epsilon(k) = \frac{1}{k}$ be the step size at the kth iteration. The distributed risk-aware real-time dispatch algorithm is given in Algorithm 4. Note that in some cases, the optimal solutions $\hat{u}_i^*$ may not equal to u_i for some EVs. Therefore, for these EVs, the unsatisfied amount of power, $(u_i - \hat{u}_i^*)$, is purchased from the real-time market, and these EVs can get charged in any slot within its charging period to compensate $(u_i - \hat{u}_i^*)$. In this chapter, we choose the slots closest to the plug-in time for these EVs. Obviously, various other selection methods can be devised, which can be used to provide regulation services to the power system [87].

Remarks: Algorithm 4 treats the aggregator and each EV as *processors in a distributed computation system* to solve the dual problem (4.12), which can significantly reduce the computational complexity of the aggregator, when there are thousands of EVs in the system.

Algorithm 4 Distributed risk-aware real-time dispatch

1) *Aggregator's algorithm:* The aggregator updates the shadow prices for each slot and broadcasts to all EVs in each iteration.

$$\nu_t(k+1) = \left(\nu_t(k) + \epsilon(k)(\textstyle\sum_i p_i^t(k) - P_t^{da})\right)^+. \tag{4.13}$$

2) *EV i's algorithm:* Given the shadow prices in each slot, each EV updates its charging strategy and feedbacks to the aggregator in each iteration.

$$p_i^t(k) = \begin{cases} p_i^{\max}, & c_t^{da} - c_p - \lambda_i(k) + \nu_t(k) < 0 \\ 0, & \text{otherwise} \end{cases}, \tag{4.14}$$

$$\hat{u}_i(k) = \begin{cases} u_i, & c_r - \lambda_i(k) > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{4.15}$$

$$\lambda_i(k+1) = \left(\lambda_i(k) + \epsilon(k)(\hat{u}_i - \textstyle\sum_{t=a_i}^{d_i} p_i^t(k))\right)^+. \tag{4.16}$$

Proposition 4.3.2. *The distributed risk-aware real-time dispatch algorithm converges to an optimal solution to (4.10), as k goes to infinity.*

Proof. The proof is given in Appendix 4.6.2. □

4.4 Performance Evaluation

In this section, we evaluate the performance of our risk-aware day-ahead scheduling algorithm based on real data, and compare it against a risk-oblivious day-ahead scheduling algorithm, which only takes into account the average energy demands of EVs. The EV specifications are obtained from typical EV specifications [88], and we assume $p_i^{\max} = 1.92$ kW for all EVs. The day-ahead prices are obtained from ISO New England on March 6, 2012 [89]. From the NHTS dataset [86], each EV may travel with different purposes everyday, including work, family errands, social and recreational purposes. For the purpose of illustrating the influence of EVs' random energy demands on the day-ahead scheduling and real-time dispatch, the random driving activities of

EVs are simplified as *i.i.d.* Bernoulli random variables in the simulation, i.e., $u_i = u^{\min}$ with probability q, and $u_i = u^{\max}$ with probability $1 - q$.

Fig. 4.1 compares the day-ahead scheduled power provided by Algorithm 3 with the risk-oblivious day-ahead scheduling algorithm, which greedily minimizes the charging cost by scheduling power over the interval with the lowest day-ahead prices, and ignores the potential scheduling risks discussed in Section 4.2.3. As illustrated in Fig. 4.1, our risk-aware scheduling algorithm strikes a balance between the charging cost and the scheduling risk by scheduling power over the intervals with comparatively higher day-ahead prices, which reduces not only the overall charging cost due to the scheduling risk, but also the peak demand of EV charging.

Fig. 4.2 illustrates the average charging cost reduction under different q and c_r. The average charging cost reduction is defined as the ratio of the average cost difference between the risk-aware and risk-oblivious algorithms to the average cost given by the risk-oblivious algorithm. As shown in Fig. 4.2, in general, the average charging cost reduction increases with q (the randomness of EVs' demands) and c_r (the under-scheduling cost). When $q = 0$, i.e., EVs' demands are deterministic, our risk-aware scheduling algorithm degenerates into the risk-oblivious day-ahead scheduling algorithm.

4.5 Conclusion

This chapter studied risk-aware day-ahead scheduling and real-time dispatch for plug-in electric vehicles. We proposed the risk-aware day-ahead scheduling algorithm that minimizes the EV charging cost and the risk of the load mismatch between the forecasted and the actual EV loads, due to the ran-

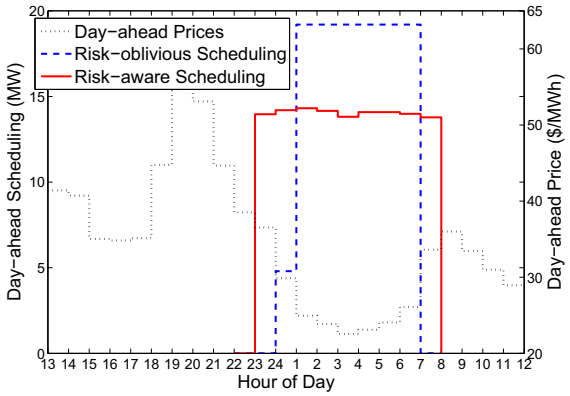

Figure 4.1: Risk-aware day-ahead scheduling vs. risk-oblivious day-ahead scheduling. $u^{\min} = 8$ kWh, $u^{\max} = 16$ kWh, $q = 0.5$, and $c_r = 63$ \$/MWh.

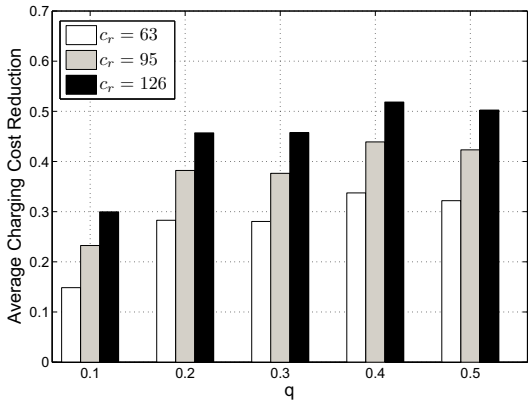

Figure 4.2: Average charging cost reduction of the risk-aware day-ahead scheduling.

dom driving activities of EVs. Although the risk-aware day-ahead scheduling problem is nonconvex, by utilizing the *hidden convexity* structure, it can be recast as a two-stage stochastic linear program, which can be solved by using the L-shaped method. Further, the distributed risk-aware real-time dispatch algorithm was developed, where the aggregator only needs to compute the shadow prices for each EV to optimize its own charging strategy in a distributed manner. Based on real data, the simulation results showed that the

proposed risk-aware day-ahead scheduling algorithm can reduce not only the overall charging cost, but also the peak demand of EV charging.

4.6 Appendix

4.6.1 Proof of Proposition 4.3.1

The objective function of (4.2) can be rewritten as

$$\begin{aligned}\sum_{t=1}^{T} Cost(t) = & \underbrace{\sum_{t=1}^{T} E_u\{c_r(P_t - P_t^{da})^+\}}_{\text{Energy purchased from real-time market}} \\ & + \underbrace{\sum_{t=1}^{T} E_u\{c_p(P_t^{da} - P_t)^+ + c_t^{da}\min(P_t, P_t^{da})\}}_{\text{Energy purchased from day-ahead market}}.\end{aligned} \tag{4.17}$$

The first term in (4.17) corresponds to the cost of purchasing energy from real-time market. Given the EVs' demand u, the charging strategy of each EV can always be divided into two parts according to the energy source. Let $\{p_i^t(u)\}$ be the charging strategy of EV i. We can divide $\{p_i^t(u)\}$ into $\{p_i^t(u)\}^{\mathrm{R}}$ and $\{p_i^t(u)\}^{\mathrm{D}}$, where $\{p_i^t(u)\}^{\mathrm{R}}$ and $\{p_i^t(u)\}^{\mathrm{D}}$ correspond to the energy purchased from the real-time market and the day-ahead market, respectively, and $\{p_i^t(u)\} = \{p_i^t(u)\}^{\mathrm{R}} \bigcup \{p_i^t(u)\}^{\mathrm{D}}$. Abusing notation slightly, let $p_i^t(u)$ denote the energy purchased from the day-ahead market. Therefore, for EV i, $\sum_t p_i^t(u)$ may be less than its demand u_i, if the day-ahead schedule is insufficient at some slot t. The unsatisfied demand is then given by $u_i - \sum_{t=a_i}^{d_i} p_i^t(u)$. Therefore, the total cost of purchasing energy from real-time market can be written as $\sum_i c_p(u_i - \sum_{t=a_i}^{d_i} p_i^t(u))$, which equals to the first term in (4.17) when the EVs' demand is u. Hence, with additional constraints $\sum_t p_i^t(u) \leq u_i,\ \forall\ i, u$

and $\sum_i p_i^t(u) \leq P_t^{da},\ \forall\ t, u$, (4.17) can be rewritten as

$$\begin{aligned}
\textstyle\sum_{t=1}^{T} Cost(t) &= E_u\{\textstyle\sum_i c_p(u_i - \sum_{t=a_i}^{d_i} p_i^t(u))\} \\
&\quad + \textstyle\sum_{t=1}^{T} E_u\{c_p(P_t^{da} - \sum_i p_i^t(u)) + c_t^{da} \sum_i p_i^t(u)\} \\
&= \textstyle\sum_i c_p E_u\{u_i\} + \sum_t c_p P_t^{da} \\
&\quad + E_u\{\textstyle\sum_{t,i}(c_t^{da} - c_p - c_r) p_i^t(u)\}.
\end{aligned} \tag{4.18}$$

Since $\sum_i c_p E_u\{u_i\}$ is constant, therefore, Proposition 4.3.1 follows.

4.6.2 Proof of Proposition 4.3.2

Let $\nu(k) = (\nu_1(k), \cdots, \nu_T(k))'$ and $\lambda(k) = (\lambda_1(k), \cdots, \lambda_N(k))'$ denote the column vectors for the dual variables, where x' denotes the transpose of the vector x. Accordingly, let $s^\nu(k)$ and $s^\lambda(k)$ denote the column vectors for the subgradient of the dual function (4.11), where the tth item in $s^\nu(k)$ and the ith item $s^\lambda(k)$ are $(\sum_i p_i^t(k) - P_t^{da})$ and $(\hat{u}_i - \sum_{t=a_i}^{d_i} p_i^t(k))$ respectively. To simplify the notation, define $\mu(k) = (\nu(k)', \lambda(k)')'$ and $s(k) = (s^\nu(k)', s^\lambda(k)')'$. We use $||\mu(k)||$ to denote the standard Euclidean norm, $||\mu(k)|| = (\mu(k)'\mu(k))^{\frac{1}{2}}$.

Let μ^* denote the dual optimal solutions to (4.12). By using (4.13) and (4.16), we have

$$\begin{aligned}
||\mu(k+1) - \mu^*||^2 &= ||(\mu(k) + \epsilon(k)s(k))^+ - \mu^*||^2 \\
&\leq ||\mu(k) + \epsilon(k)s(k) - \mu^*||^2 \\
&= ||\mu(k) - \mu^*||^2 + (\epsilon(k))^2||s(k)||^2 \\
&\quad -2\epsilon(k)s(k)'(\mu(k) - \mu^*) \\
&\leq ||\mu(k) - \mu^*||^2 + (\epsilon(k))^2||s(k)||^2 \\
&\quad +2\epsilon(k)(D(\mu(k)) - D^*),
\end{aligned} \tag{4.19}$$

where $D^* = D(\mu^*)$ is the dual optimal value and $D(\cdot)$ is the dual function defined in (4.11). The last line of (4.19) follows from the definition of subgradient, which gives $D^* \leq D(\mu(k)) + s(k)'(\mu(k) - \mu^*)$.

Applying the inequality (4.19) recursively, we have

$$\begin{aligned} ||\mu(k+1)-\mu^*||^2 \;\leq\; & ||\mu(1)-\mu^*||^2+\textstyle\sum_{i=1}^{k}(\epsilon(i))^2||s(i)||^2 \\ & +2\textstyle\sum_{i=1}^{k}\epsilon(i)(D(\mu(i))-D^*). \end{aligned} \tag{4.20}$$

Assume that (4.9) has a strictly feasible solution, i.e., the Slater condition holds in (4.9).[2] By using the fact that for any μ, $D(\mu)$ is a lower bound of the primal problem (4.9), it can be shown that $||\mu(1)-\mu^*||$ is upper bounded by a constant $B>0$. Since $||\mu(k+1)-\mu^*||^2 \geq 0$, we have

$$2\textstyle\sum_{i=1}^{k}\epsilon(i)(D^*-D(\mu(i))) \leq B^2+\sum_{i=1}^{k}(\epsilon(i))^2||s(i)||^2. \tag{4.21}$$

Let $\Delta(k)=\min_{i=1,\cdots,k}(D^*-D(\mu(i)))$ denote the smallest distance between the dual optimal value and the best dual value given by the k iterations. Combining $\sum_{i=1}^{k}\epsilon(i)\Delta(k) \leq \sum_{i=1}^{k}\epsilon(i)(D^*-D(\mu(i)))$ with (4.21), we have

$$\Delta(k) \leq \frac{B^2+\sum_{i=1}^{k}(\epsilon(i))^2||s(i)||^2}{2\sum_{i=1}^{k}\epsilon(i)}. \tag{4.22}$$

Since in each iteration $p_i^t(k)$ and $\hat{u}_i(k)$ for any t and i are bounded, the subgradient $s(k)$, the affine function of $p_i^t(k)$ and $\hat{U}_i(k)$, is also bounded. Let $G>0$ denote the upper bound of $s(k)$, i.e., $||s(i)||^2 \leq G^2$. We have

$$\Delta(k) \leq \frac{B^2+G^2\sum_{i=1}^{k}(\epsilon(i))^2}{2\sum_{i=1}^{k}\epsilon(i)}. \tag{4.23}$$

Since $\epsilon(k)=\frac{1}{k}$, $\lim_{k\to\infty}\sum_{i=1}^{k}(\epsilon(i))^2<\infty$ and $\lim_{k\to\infty}\sum_{i=1}^{k}\epsilon(i)=\infty$. Therefore, Proposition 4.3.2 follows.

[2]This assumption holds, if $P_t^{da}>0$ for some t.

Chapter 5

STOCHASTIC OPTIMIZATION BASED ECONOMIC DISPATCH AND INTERRUPTIBLE LOAD MANAGEMENT WITH INCREASED WIND PENETRATION

5.1 Introduction

In order to meet the renewable portfolio standards (RPS) adopted by many states in the U.S., much effort is being made to integrate renewable generation (particularly wind generation) into bulk power grids. Indeed, wind energy constitutes a significant portion of this renewable integration [3]. High penetration of wind generation, however, is expected to result in significant operational challenges [4], due to its non-dispatchability and variability. Accurate forecasts of future wind generation across temporal and spatial scales remains still elusive [5]. Therefore, the integration of wind generation at high penetration levels into bulk power grids may have significant impact on system reliability, because of the inability to obtain an acceptable load/generation balance.

One possible approach to integrate wind generation into power system operational planning is to treat wind generation as negative load [90]. Conventionally, in power system operations, the effects of load forecast errors are mitigated by the regulation and operating reserves that are co-scheduled with generation. However, as pointed out in [91], wind generation is much more variable and unpredictable than the load. It is clear that, at a high penetration level, wind generation would become a dominating factor in terms of uncertainty; and hence there is an urgent need to revisit the approach to scheduling

regulation and operating reserves (see, e.g., recent studies [92, 93] on this issue). Further, it is reasonable to expect that the cost of regulation services and operating reserves could increase significantly when the penetration level of wind generation is high.

Aiming to maintain system reliability with high penetration of wind generation and reduce the cost of reserves, joint optimization of economic dispatch (ED) and interruptible load management is explored in this chapter. Interruptible loads have been recognized as one of the ancillary services, particularly, as a contingency reserve service. In an interruptible load program, the customer enters into a contract with the independent system operator (ISO) to reduce its demand when requested. The ISO benefits by reducing its peak load and thereby saving costly reserves, restoring quality of service and ensuring reliability, and the customer benefits from the reduction in energy costs and from the incentives provided by the contract. By using interruptible load management, the system may save costly reserves for load/generation imbalance due to wind generation forecast errors.

In addition to interruptible load management, accurate wind generation forecast is expected to reduce the requirement of regulation services and operating reserves. A vast amount of the existing literature on wind generation forecast focuses on wind speed forecast which is subsequently translated into the wind power output based on the turbine power curve. In fact, as shown in [94], the power outputs from identical turbines within a farm are not necessarily equal, even if the turbines are co-located. This "mismatch" is more severe when they are far apart, and could result in an erroneous forecast of wind farm output. In this work, the Markov-chain-based wind farm generation forecast is leveraged to improve the forecast accuracy.

Based on recent work [94, 95], a novel ED model is exploited by leveraging the Markov-chain spatio-temporal wind farm generation forecast and interruptible load management, aiming to minimize the system operating costs. In light of the Markovian property of the proposed forecast model, the ED problem is formulated as a Markov decision process based dynamic programming problem, which, however, is often not tractable due to the "curse of dimensionality." To reduce the complexity of this joint optimization of ED and interruptible load management, the "greedy" policy [39] is used, which minimizes the immediate system cost only. Utilizing the convexity of the optimization problem, the corresponding ED problem is further transformed into a stochastic program by using the point forecast of wind generation. Worth noting is that, with the high forecast accuracy of the spatio-temporal wind farm generation, it turns out that this transformation can reduce the system cost, which is verified by simulations. By using actual wind power data, the proposed ED is tested on the IEEE Reliability Test System – 1996 [96]. Such a simulation study is carried out for different wind penetration levels so as to demonstrate economic benefits of the proposed ED framework.

5.1.1 Related Work

A significant body of work in the literature indicates that operating reserves could be procured from demand response [97], instead of from backup generation capacity. Black *et al* [98] propose a methodology for integrating demand response into optimal dispatch algorithms for electric power systems, which takes into account the impact of load shifting to later time periods. The work by Papavasiliou *et al* [99] presents a contract for integrating renewable energy supply and electricity spot markets for serving deferrable electric loads in order to mitigate renewable energy intermittency. In [100], three different method-

ologies are proposed to integrate short-term responsiveness into a generation technology mix optimization model while considering operational constraints. Recent studies [101] and [102] use several case studies to demonstrate that real-time pricing could help improve the usage of wind generation and have the potential to reduce the cost brought by wind generation integration. Unfortunately, a lack of real-time pricing has prevented most consumers from seeing and responding to real-time prices, resulting in inelastic demand in the short-term [103]. In this chapter, interruptible load management is integrated and jointly optimized with ED.

Most existing works on wind power integration utilize scenario-based stochastic programming to address the uncertainty and variability of wind power [90,92,104]. Papavasiliou *et al* [105] analyze the reserve requirements for wind power integration based on a two-stage stochastic programming model. Restrepo *et al* [106] assess the yearly impact of wind power through a hybrid deterministic/stochastic unit commitment. Constantinescu *et al* [107] propose a framework for integrating a numerical weather prediction model in stochastic unit commitment/economic dispatch formulations that account for wind power uncertainty.

There is very limited research on the impact of wind forecast errors on the real-time dispatch in market operation. The forecast errors, which are the "mismatches" between what is scheduled in the ED stage and the real-time dispatch, may put forth great challenge for system operators to balance the unexpected surplus or deficit of wind power. Wang *et al* [108] investigate the impact of wind power forecasting on unit commitment by using the realized wind generation in ED. In general, even in the ED stage, perfect wind generation forecasts are unobtainable due to the variability of wind and thus wind

forecast errors also exist in ED. Therefore, one focus of this chapter is on the impact of wind forecast errors on the real-time dispatch in market operation.

The rest of the chapter is organized as follows. In Section 5.2, the ED model with interruptible load management is proposed by leveraging the spatio-temporal wind forecast. In Section 5.3, an illustrative power system ED example is presented, which quantifies the potential savings in both generation cost and regulation reserves in the proposed ED model. The chapter is concluded in Section 5.4.

5.2 Economic Dispatch with Short-term Wind Generation Forecast

In this section, the economic dispatch (ED) problem with short-term wind farm generation forecast is formulated, with focus on the impact of wind generation on system operation. Worth noting is that load uncertainty and forced outages of generators and transmission lines are not directly considered. The key notation used in ED is summarized in Table 5.1.

5.2.1 Spatio-temporal Wind Farm Generation Forecast

Due to the non-dispatchability and variability of wind generation, efficient integration of wind energy in power grids is challenging, and lies on accurate forecasting models. A vast amount of existing literature on wind generation forecast focuses on wind speed forecast, assuming that wind generation from the farm can be directly calculated as a function of wind speed recorded at one specific location in the farm. In reality, however, even if the turbines are co-located, the power outputs from identical turbines within a farm are not necessarily equal. This "mismatch" gets more severe when they are far apart. Therefore, forecast errors of the existing approaches can be large, and can

Table 5.1: Notation used in economic dispatch.

b	bus index
g	generator index
w	wind farm index
B	set of buses
$C_b^I(\cdot)$	interruptible load cost function at bus b
$C_g^G(\cdot)$	generation cost function of generator g
$C_g^R(\cdot)$	reserve cost function of generator g
D_b^t	load at bus b at time t
G	set of generators
L_b^t	load at bus b that could be interrupted at time t
$P_{bb'}$	branch power flow
$P_{bb'}^{\max}$	rated capacity of transmission line bb'
P_b^t	scheduled interruptible load at bus b at time t
P_g^t	scheduled power output of generator g at time t
$P_g^{\min}, P_g^{\max}$	minimal/maximal power output of generator g
P_w^t	scheduled generation from the wind farm w at time t
$P_w^{\min}, P_w^{\max}$	minimal/maximal generation from wind farm w
$\hat{P}_w^t$	"point-forecast" of the generation from wind farm w
R_g^t	scheduled regulation reserve from generator g at time t
R_s	regulation reserve requirement
RP_g^{up}, RP_g^{dn}	maximal ramping up/down of generation g
V_b	voltage of bus b
$V_b^{\max}, V_b^{\min}$	upper and lower voltage limits of bus b
W	set of wind farms
$Y_{bb'}$	element of network admittance matrix
$\theta_{bb'}$	angle associated with $Y_{bb'}$
δ_b	voltage angle of bus b

impact the ED results.

In this study, a spatio-temporal analysis approach for wind generation forecast is considered (which is proposed in the companion paper of this work [95]), while taking into account the diurnal non-stationarity and the seasonality of wind speed. Specifically, a graphical model to capture the spatial correlation between the power outputs from the wind turbines is developed, i.e., a minimum spanning tree is constructed based on graph theory. The spatial correlation between the individual wind turbines is determined by using a

linear regression model. Based on these steps, the probability distribution F_w of farm aggregate wind generation can be characterized using the wind speed measured at the reference meteorological tower in the farm.

Next, the temporal correlation is analyzed by using a finite state Markov chain model. A key objective is to capture the statistical distribution and temporal dynamics of aggregate wind farm generation $\mathcal{P}_w$ by using a Markov chain with the following characteristics: 1) the Markov chain has N_s (N_s is finite) states. Let $\mathcal{S}$ denote the state space of the Markov chain. Specifically, state $S_k = [\Gamma_k, \Gamma_{k+1})$, $k \in \{1 \dots N_s\}$, with extreme values given by $\Gamma_1 = 0$ and $\Gamma_{N_s+1} = P_w^{\max}$; 2) the Markov chain is discrete-time and of order 1. These characteristics are imposed to make the forecast model amenable to practical applications, but this can be generalized by using higher-order Markov chains (the approach developed in [61] is followed in the chapter). Define the quantity τ_k as the average duration that $\mathcal{P}_w$ stays in state S_k,

$$\tau_k = \frac{F_w(\Gamma_{k+1}) - F_w(\Gamma_k)}{(\mathcal{N}(\Gamma_k) + \mathcal{N}(\Gamma_{k+1}))}, \tag{5.1}$$

where $\mathcal{N}(\Gamma)$ denotes the level crossing rate (the number of times per unit time that the farm aggregate power process $\mathcal{P}_w$ crosses Γ) for the farm aggregate power Γ ($\Gamma \geq 0$) [61].

A key observation from (5.1) is that for a smaller value of τ_k, the random process $\mathcal{P}_w$ is more likely to switch out of the state S_k within a time slot[1], *and hence the random process would not be sufficiently captured by the corresponding discrete-time Markov chain.* Therefore, using the method in [61], we introduce a constant τ to find the $N_s - 1$ variables $\{\Gamma_2, \Gamma_3, \dots \Gamma_{N_s}\}$, i.e., solving (5.1) with $\tau_k = \tau$, $\forall k \in \{1 \dots N_s - 1\}$. Then, the transition

[1]The length of a time slot is the same as the interval proposed for the ED. In the chapter, the length of a time slot is 10 minutes.

probability matrix could be easily obtained by using the data, as follows:

$$[Q]_{i,j} = \frac{n_{ij}}{\sum_{k=1}^{N_s} n_{ik}}, \tag{5.2}$$

where n_{ij} is the number of transitions from S_i to S_j.

Note that due to the diurnal non-stationarity and the seasonality of wind speed, the Markov chain is non-stationary. The seasonality can be tackled by designing the forecast model for each month individually. The diurnal non-stationarity can be handled by identifying an epoch T_d such that the wind generation exhibits stationary behavior within every T_d-length epoch and designing a forecast model for each of these T_d-length epochs separately. For the data at hand, $T_d = 3$ hours appears to be a reasonable choice.

5.2.2 Interruptible Load Management

The forecast errors are inevitable, which would incur complications in system operations, since the realized wind generation may be different from the forecast. When the realized wind generation is larger than the forecast, the wind generation may have to be curtailed if sufficient downward reserves from other resources are not present. When the realized wind generation is lower than the forecast, the system needs to compensate for the forecast errors by using regulation reserves (e.g., spinning reserves). Clearly, the required reserves increase with the penetration level of wind. Therefore, the reserve cost can be high when the penetration level of wind is high. To reduce the cost, interruptible load services can be used when the realized wind generation is lower than the forecast. The ISO can reduce the demand and thereby withhold costly reserves. In this work, it is assumed that the interruptible load contracts are given a priori, i.e., the amount of load that can be interrupted in a given period

T_{IL} is known [109, 110].[2]

5.2.3 Problem Formulation: A Markov Decision Process View

In this section, the ED problem is formulated to take into account both spatio-temporal wind generation forecast and interruptible load services. Specifically, leveraging the Markov-chain-based forecast model, the ED problem is cast as a Markov decision process (**P1**), and two forecasting approaches are compared in the study: one with all possible forecasted wind farm generation states (**P2**), and the other with the point forecast of wind farm generation (**P3**). As expected, in the presence of the forecast errors, the performance of these two approaches depends on the forecast accuracy.

Horizon: The number of slots within the period T_{IL} is the time horizon. Let T denote the total number of slots in the period T_{IL}.

State: At slot t, the wind generation state S_w^t of wind farm $w \in W$ and the remaining interruptible load L_b^t at bus $b \in B$ before the end of the period T_{IL} is defined as the system state, $S^t = \{\{S_w^t\}_{w\in W}, \{L_b^t\}_{b\in B}\}$. Note that the state of wind S_w^t is observed based on the realized wind generation in the previous slot, and determines the possible wind generation states, i.e., the forecasted wind generation state space $\mathcal{S}_w^t$ in the current slot according to the Markov chain model.

Action: Given the system state, the action is defined as the scheduled power generation of each conventional generator $P_g^t(s)$, the scheduled wind generation of each wind farm $P_w^t(s)$, the amount of interruptible load to be cut off on each load bus $P_b^t(s)$, and the scheduled reserve of each conventional generator $R_g^t(s)$ for each scenario $s \in \prod_{w\in W} \mathcal{S}_w^t$.[3] Note that the action in each

[2] The time scale of some types of contracts is large, e.g., one month in [109].

[3] $\prod_{w\in W} \mathcal{S}_w^t$ denotes the Cartesian product of all $\mathcal{S}_w^t$.

slot t is subject to the balance constraint,

$$\sum_{g\in G} P_g^t(s) + \sum_{w\in W} P_w^t(s) = \sum_{b\in B} D_b^t - \sum_{b\in B} P_b^t(s), \tag{5.3}$$

and the system reserve requirement constraint,

$$\sum_{g\in G} R_g^t(s) \geq R_s. \tag{5.4}$$

Policy: The policy π is a sequence of mappings from states to actions, which define a controller that takes actions as specified by the policy.

Given the policy π, the expected system cost can be computed from slot t to T by the following recursive function:

$$V_t^\pi(S^t) = cost(S^t, \pi) + E_{S^t}^\pi\{V_{t+1}^\pi(S^{t+1})\}, \tag{5.5}$$

where $cost(S^t, \pi)$ denotes the expected system cost at slot t, given by

$$\begin{aligned} cost(S^t, \pi) = E_{S^t}^\pi\Big\{ &\sum_{g\in G} C_g^G(P_g^t(s)) + \sum_{b\in B} C_b^B(P_b^t(s)) \\ &+ \sum_{g\in G} C_g^R(R_g^t(s))\Big\}, \end{aligned} \tag{5.6}$$

where the last term denotes the cost of providing reserve services. Therefore, the ED problem in the form of a Markov decision process based dynamic programming is given by

$$\textbf{P1}\text{: } \min_\pi \ cost(S^t, \pi) + E_{S^t}^\pi\{V_{t+1}^\pi(S^{t+1})\}, \tag{5.7}$$

with the system and individual units operating constraints to be elaborated later in the formulation **P3**.[4]

The objective of (5.7) is to find the optimal policy π^* that minimizes the expected system cost (5.5). One main challenge of solving the above

[4]The constraints in **P1** are scenario-based and essentially the same as those in **P3**.

dynamic programming problem is the curse of dimensionality, i.e., the number of states grows exponentially with the horizon, making it impossible to solve the problem at every slot (with a length of several minutes) even for moderate size systems. In this work, the greedy policy is applied to solve the problem (5.5), i.e., in each slot, only the immediate system cost (5.6) is minimized without considering the system cost in the future, i.e.,

$$\mathbf{P2}\text{: } \min_{\pi} \ cost(S^t, \pi). \tag{5.8}$$

Observing that (5.6) is convex[5], the objective function of **P2** can be lower bounded by using Jensen's inequality:

$$\begin{aligned} cost(S^t, \pi) \geq & \sum_{g\in G} C_g^G(E_{S^t}\{P_g^t(s)\}) \\ & + \sum_{b\in B} C_b^I(E_{S^t}\{P_b^t(s)\}) + \sum_{g\in G} C_g^R(E_{S^t}\{R_g^t(s)\}). \end{aligned} \tag{5.9}$$

When the constraints (e.g., (5.3) and (5.4)) are relaxed by using their expectations, the ED problem can be recast as a stochastic program with point wind generation forecast[6] (this transformation will be further elaborated later):

$$\mathbf{P3}\text{: } \min \ \sum_{g\in G} C_g^G(P_g^t) + \sum_{b\in B} C_b^I(P_b^t) + \sum_{g\in G} C_g^R(R_g^t) \tag{5.10}$$

subject to

$$\sum_{g\in G} P_g^t + \sum_{w\in W} P_w^t = \sum_{b\in B} D_b^t - \sum_{b\in B} P_b^t \tag{5.11}$$

$$\begin{aligned} & \sum_{g\in G_b} P_g^t + \sum_{w\in W_b} P_w^t - D_b^t + P_b^t \\ & = \sum_{b'} |V_b||V_{b'}||Y_{bb'}| \cos(\theta_{bb'} + \delta_b - \delta_{b'}), \forall b \in B \end{aligned} \tag{5.12}$$

$$V_b^{\min} \leq |V_b| \leq V_b^{\max}, \forall b \in B \tag{5.13}$$

[5]The functions C_g^G, C_b^B and C_g^R are convex.

[6]Based on (5.9), the expectation can be taken with respect to each variable. Therefore, the notation is simplified in **P3**, e.g., $P_g^t = E_{S^t}\{P_g^t(s)\}$.

$$|P_{bb'}| \leq P_{bb'}^{\max}, \forall b, b' \in B, b \neq b' \tag{5.14}$$

$$\sum_{g \in G} R_g^t \geq R_s \tag{5.15}$$

$$-RP_g^{dn} \leq P_g^t - P_g^{t-1} \leq RP_g^{up}, \forall g \in G \tag{5.16}$$

$$P_g^{\min} \leq P_g^t + R_g^t \leq P_g^{\max}, \forall g \in G \tag{5.17}$$

$$0 \leq P_b^t \leq L_b^t, \forall b \in B \tag{5.18}$$

$$P_w^{\min} \leq P_w^t \leq P_w^{\max}, \forall w \in W \tag{5.19}$$

$$P_w^t \leq \hat{P}_w^t, \forall w \in W \tag{5.20}$$

where

- (5.10) is the system operating costs including costs of generation, costs of using interruptible load services, and costs of providing reserve services.
- (5.11) is the system-wide power balance between the sum of the scheduled generation and the net demand due to the use of interruptible load services.
- (5.12) is the active power flow equation at bus b with the load interruption P_b^t as requested by ISO. For the sake of the brevity, the reactive power flow is not explicitly shown in the formulation.
- (5.13) is the upper and lower limit of bus voltage magnitude.
- (5.14) is the rated capacity of transmission line bb'.
- (5.15) is the system reserve requirement, which can be determined by the system reliability requirement, the wind forecast accuracy and load forecast accuracy.
- (5.16) is the ramping constraint of each conventional generator.

- (5.17) is the capacity constraint of each generator for providing reserve services.

- (5.19) is the upper and lower bound of wind farm output.

- (5.20) is the point wind generation forecast for each wind farm, where $\hat{P}_w^t$ is the wind generation given by the forecast models. Given the transition matrix of the Markov chain Q_w of wind farm w and the most recent aggregate wind generation at time $t - t_l$,[7] $\hat{P}_w^t$ is given by:

$$\hat{P}_w^t = \sum_{k=1}^{N_s} p_k [Q_w^{t_l}]_{i_0,k}, \tag{5.21}$$

  where i_0 denotes the state of aggregate wind generation at time $t - t_l$, and p_k is the representative generation level of state S_k.

Remarks:

- Note that **P3** uses only the conditional expected wind generation of the proposed forecast model. Therefore, the computational complexity of solving **P3** is far less than that of solving **P2**, which takes into account all possible states.

- It can be shown that the constraints in **P3** are less stringent than those in **P2**, in the sense that any feasible solution to **P2** would also satisfy **P3**. For example, given any feasible solution to **P2** that satisfies (5.3), it also satisfies (5.11) after taking expectation. This implies that the optimal value of **P3** is no greater than that of **P2**. Note that, in this chapter, the forecast errors are accounted for, which indicates that the forecasted scenarios may be different from the realized wind generation. Hence, the

[7] Here, t_l is used to characterize the "forecast lead time", e.g., $t_l = 1$ for 10-min forecast and $t_l = 6$ for 1-hour forecast.

real system cost highly depends on the forecast accuracy. Worth noting is that the conditional expected wind generation (point wind generation forecast) is close to the realized wind generation (e.g., Fig. 5.1). It turns out that this transformation would decrease the system cost, as illustrated in Section 5.3.

- The regulation reserves maintained in the real-time dispatch are assumed to be adequate to accommodate the wind generation uncertainty.[8]

5.3 Case Studies

5.3.1 Data and Simulation Setup

In this section, the proposed ED formulation is applied to the IEEE Reliability Test System (RTS) – 1996 [96] under different wind penetration levels to simulate the impact of using different wind generation forecasts on the system. The simulation period is 24 hours. Before performing ED, unit commitment (UC) is run to determine which generators should be "on" during a 24-hour horizon, where typical technical restrictions are accounted for, such as the minimum up and down time limits and the startup costs of generators [111],[9] and the wind generation of each wind farm is estimated from historical data [94].[10] Then, ED is run every 10 minutes for 24 hours. In both UC and ED, the "3+5" rule is used to determine the regulation reserve requirement R_s, which equals 3% of hourly forecast load plus 5% of hourly forecast wind power [112]. It turns out that thanks to the significant gain, the proposed ED together with the

[8] In general, the available ramping capability of on-line units in the system may not be sufficient to handle unforeseen decreases in wind generation, which would impact the reliability of the system.

[9] Specifically, a cold startup cost and a warm startup cost were considered, depending on the length of time that the unit is down.

[10] For the available data, the data of Jan. 2009 is used for spatio-temporal analysis and Markov chain design.

Table 5.2: Hourly load (Unit: MW).

HR1	HR2	HR3	HR4	HR5	HR6	HR7	HR8
5729	5387	5130	5045	5045	5130	6327	7353
HR9	HR10	HR11	HR12	HR13	HR14	HR15	HR16
8123	8208	8208	8123	8123	8123	7952	8038
HR17	HR18	HR19	HR20	HR21	HR22	HR23	HR24
8465	8550	8550	8208	7781	7097	6242	5387

"3+5" rule is adequate to accommodate the wind generation uncertainty in the simulations.

The hourly profile of the loads is taken from [96] with a peak value 8550 MW, as shown in Table 5.2. In the RTS-1996 system, the large coal-fired plant U350 is assumed to be replaced by the wind farm, and the wind generation data on Jan. 7th, 2010 [94] is used after proper scaling to suit the chosen penetration level. The characteristics of the power plants (as shown in Table 5.3) are obtained from [96], where the startup costs of the power plants are computed based on the fuel cost presented in Table 5.5 [113]. The generation cost functions of the power plants are assumed to be quadratic functions with coefficients given in Table 5.4 [114]. The reserve cost functions and the interruptible load cost functions are assumed to be linear, with coefficients 50$/MWh and 20$/MWh, respectively.[11]

For the current setup, Table 5.6 summarizes the results of the UC under different wind penetration levels. For ease of comparison, the rows show the number of units "on" by generator type, where the upper row corresponds to the case with 10% wind integration, the middle row corresponds to the case with 20% wind integration, and the lower row corresponds to the case

[11]For the sake of studying the impact of wind generation forecast on ED, the cost functions of reserve and interruptible load are simplified. The coefficients are chosen based on locational marginal prices. More complicated functions that take into account ancillary service markets can be considered, which is beyond the scope of this study.

Table 5.3: Generator parameters.

Units	Unit Type	Fuel	Min. Down Time [h]	Min. Up Time [h]	$RP_{G_i}^{up}$ ($RP_{G_i}^{dn}$) [MW/min]	Initial State
U12	Oil/Steam	#6 Oil	2	4	1	-4
U20	Oil/CT	#2 Oil	1	1	3	-1
U50	Hydro	-	-	-	-	-
U76	Coal/Steam	Coal	4	8	2	-8
U100	Oil/Steam	#6 Oil	8	8	7	-8
U155	Coal/Steam	Coal	8	8	3	-8
U197	Oil/Steam	#6 Oil	10	12	3	-12
U400	Nuclear	LWR	1	1	20	-

Table 5.4: Generator parameters.

Units	a_i [$/h]	b_i [$/MWh]	c_i [$/MW^{2}h]	$P_{G_i}^{\max}$ [MW]	$P_{G_i}^{\min}$ [MW]
U12	86.3852	56.564	0.328412	12	2.4
U20	400.6849	130	0	20	16
U50	0.001	0.001	0	50	10
U76	212.3076	16.0811	0.014142	76	15.2
U100	781.521	43.6615	0.052672	100	25
U155	382.2391	12.3883	0.008342	155	54.3
U197	832.7575	48.5804	0.00717	197	69
U400	395.3749	4.4231	0.000213	400	100

Table 5.5: Fuel costs.

#2 Oil	#6 Oil	Coal	Uranium
15.17 $/MBtu	8.40 $/MBtu	1.78 $/MBtu	0.60 $/MBtu

with 30% wind integration. For the sake of conciseness, consecutive hours with identical schedules have been lumped together. The run-of-the-river U50 units and the nuclear U400 units are assumed to be always on, while U20 units are turned off and thus not shown in the table.

Based on Table 5.6, the ED is run to compare the results of using the spatio-temporal wind generation forecast with conventional wind-speed-based

forecasts, in which the forecast of aggregate wind generation are given by

$$\hat{P}_w^t = \sum_{m=1}^{M} N_{w,m} U_{w,m}(\hat{W}_{w,m}^t) \tag{5.22}$$

where $\hat{W}_{i,m}^t$ is the forecasted wind speed at the meteorological tower of class m in wind farm w, $N_{w,m}$ is the number of wind turbines of class m, and $U_{w,m}(\cdot)$ is the estimate of the farm aggregate power in terms of the wind speed. Specifically, two existing forecast models of wind speed [115] are considered here:

1) Persistent prediction of wind speed (PPW): $\hat{W}_{w,m}^t = W_{w,m}^{t-1}$, where $W_{w,m}^{t-1}$ is the wind speed observed at time $t-1$;

2) Linear prediction of wind speed (LPW): $\hat{W}_{w,m}^t = \sum_{l=1}^{L} a_l W_{w,m}^{t-l}$, where the prediction coefficients a_l are obtained by fitting an L-order AR model to the wind speed data available at time $t - t_l$.

Fig. 5.1 depicts the point forecasts on Jan 7th, 2010 of different wind generation forecast approaches. It is observed from Fig. 5.1 that wind-speed-based forecasts could lead to artificial "spikes" that could have severe impact on the reliability of power systems if the forecast is used for ED. Since the focus of this work is on the effect of wind generation forecast errors on the simulation, planned and forced outages of generators are not considered. In the following simulations, the horizon is divided into eight 3-hour epochs due to the diurnal non-stationarity of the wind speed, and a Markov chain is obtained by spatio-temporal analysis for each epoch as discussed in Section 5.2.1.

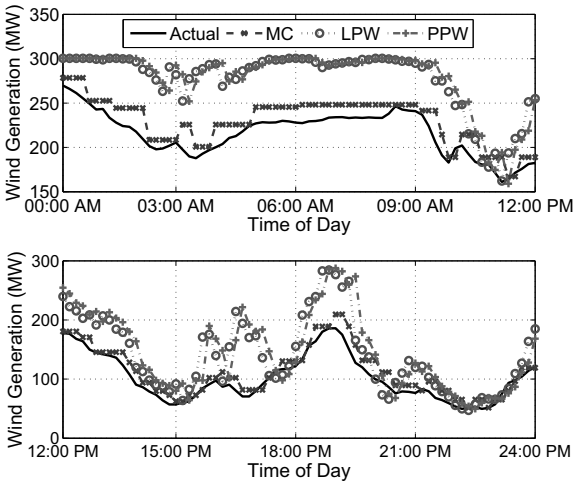

Figure 5.1: Comparison of wind generation forecast.

Table 5.6: Unit commitment decision variable.

Unit	Hour																							
	1-5	6	7	8	9	10-11		12	13	14-15		16	17	18-19		20	21	22		23	24			
U12	0											5				0								
	0					1				0			1	2			1		0					
	0		1	2		4		3		0		1	13	15		14	2		0					
U76	5		10																	9	5			
	2																			4	2			
	7															10	9		7					
U100	0											1				0		1		0				
	0					1			2			3				1			0					
	0															1			0					
U155	11					12																		
	9	10				12															11			
	3		6			12														9	6			
U197	0			3		9										5	3		0					
	0					5				4						0								
	0																							

5.3.2 Results and Discussions

Stochastic ED with Point Wind Generation Forecast

In this section, results to demonstrate the benefit of relaxing **P2** to **P3** by using the point wind generation forecast are presented. As discussed in Section

Table 5.7: Benefit of stochastic ED with point wind generation forecast (Unit: 10^5\$). The wind penetration level is 20%.

Epoch	1	2	3	4	5	6	7	8	Total
P2	1.94	1.75	2.69	3.44	3.30	3.57	3.41	2.24	22.34
P3	1.86	1.72	2.66	3.39	3.27	3.54	3.38	2.21	22.03

5.2.3, the transformation from **P2** to **P3** can reduce not only the computational complexity but also the system cost. From the simulations, the average computational times for **P2** and **P3** at each slot are 80.5s and 0.7s, respectively, where the problem is solved by using CPLEX 11.2 in Matlab 2008 on a PC with a 2.4 GHz Intel Core i3 processor and 4 GB RAM. Note that the computational time for **P2** increases exponentially with with the number of wind farms, simply because the state space increases exponentially with the number of wind farms. In the simulation, only three wind farms are considered. In practice, the computational time for **P2** can be prohibitive if more wind farms are considered.

For the system cost, Table 5.7 compares the system cost of each epoch given by **P2** and **P3**. Since the Markov-chain-based point wind generation forecast is close to the realized wind generation, the reserves needed to compensate for the forecast errors in **P3** are less than that in **P2** on average. Therefore, the system cost given by **P3** is less than that given by **P2**.

Impact of Wind Farm Generation Forecast on ED

In this section, detailed dispatch results to show the impact of different short-term wind generation forecasts are provided. An "ideal" case, which is assumed to have a perfect forecast, is used as a benchmark.

Table 5.8 summarizes the system cost given by different wind generation forecasts under different wind penetration levels, respectively. The system

Table 5.8: System cost (10^5\$) under different wind penetration levels. For each forecast case, the upper, middle and lower rows correspond to 10%, 20% and 30% wind penetration, respectively.

Epoch	1	2	3	4	5	6	7	8	Total
Ideal	2.01	1.87	3.20	4.14	4.04	4.30	4.11	2.49	26.15
	1.80	1.66	2.61	3.34	3.20	3.46	3.29	2.14	21.51
	1.64	1.49	2.15	2.46	2.39	2.69	2.71	1.96	17.49
MC	2.04	1.90	3.22	4.16	4.08	4.34	4.14	2.52	26.40
	1.86	1.72	2.66	3.39	3.27	3.54	3.38	2.21	22.04
	1.74	1.58	2.22	2.55	2.48	2.79	2.85	2.04	18.27
LPW	2.21	2.05	3.40	4.36	4.21	4.56	4.34	2.68	27.81
	2.18	2.04	3.06	3.79	3.56	4.02	3.79	2.55	24.99
	2.21	2.05	2.76	3.19	2.87	3.51	3.50	2.53	22.64
PPW	2.24	2.06	3.40	4.36	4.21	4.57	4.36	2.69	27.88
	2.25	2.04	3.05	3.80	3.54	4.04	3.84	2.57	25.12
	2.24	2.03	2.76	3.19	2.85	3.54	3.56	2.56	22.75

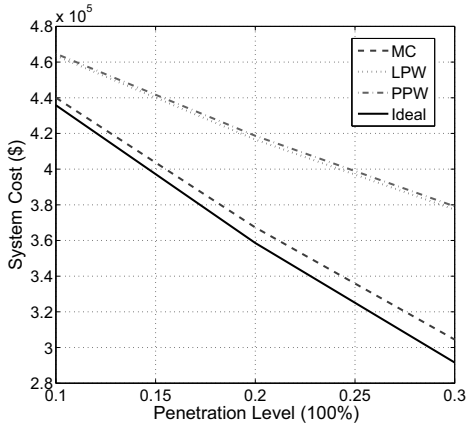

Figure 5.2: System cost vs. penetration level.

cost in each epoch is summarized and the total system cost is given in the last column. It can be observed from Table 5.8 that the system cost based on the proposed Markov chain model is close to the benchmark with perfect wind generation forecast. Compared with wind-speed-based forecasts, the system cost can be reduced by up to 20% by using the Markov-chain-based forecast. As expected, the system cost decreases with the wind penetration level, as illustrated in Fig. 5.2. However, the gaps of system cost between wind-speed-

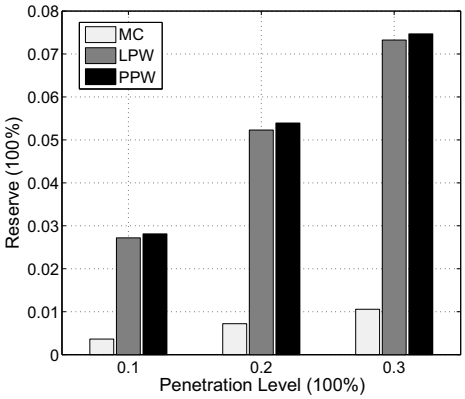

Figure 5.3: Reserve vs. penetration level.

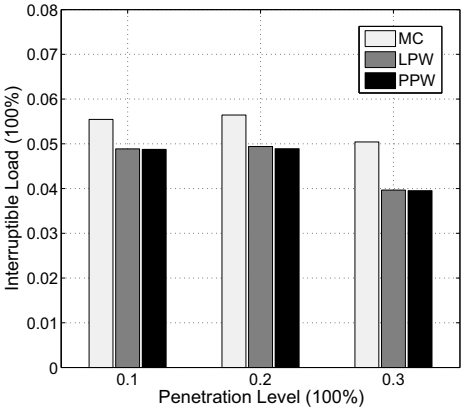

Figure 5.4: Interruptible load vs. penetration level.

based forecast models and the Markov chain forecast model increase, due to the fact that the forecast errors of wind-speed-based forecast models are amplified as the wind penetration level increases, and to compensate for the forecast errors, the system needs a large amount of reserves.

Figs. 5.3 and 5.4 illustrate the amount of regulation reserves used and the amount of load interrupted under different wind generation forecasts, respectively. In Figs. 5.3 and 5.4, reserve and interruptible load are normalized by the system load. Compared with the cases of wind-speed-based forecasts, ED with Markov-chain-based forecast requires much less reserves.

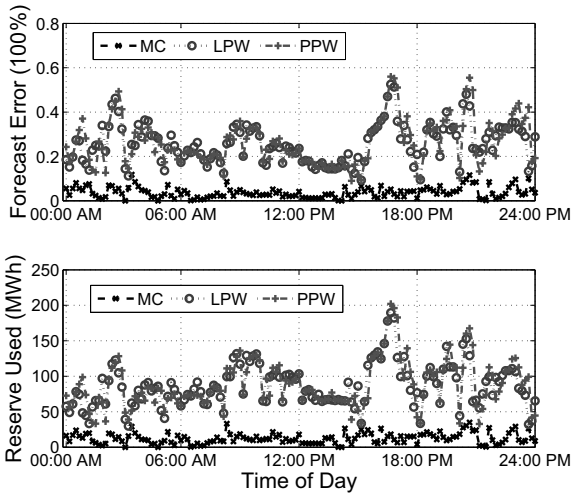

Figure 5.5: Reserve vs. estimation error. The penetration level is 30%.

When the wind penetration level is 30%, the regulation reserve requirement of the Markov-chain-based forecast is only 1% of system load, though ED with Markov-chain-based forecast would use a little more interruptible load. Since the cost of using interruptible load services is lower compared to using regulation reserves, ED with any forecast approach considers the interruptible load services. From Fig. 5.1, wind-speed-based forecasts usually overestimate the wind generation more than the Markov-chain-based forecast. Since the cost of wind generation is assumed to be zero, wind-speed-based forecasts would schedule less interruptible load than Markov-chain-based forecast.

Another key observation is that the amount of regulation reserve used is strongly correlated with the forecast error, as suggested in Fig. 5.5, where the forecast error is normalized by the realized wind generation. The curves of estimation errors and reserves follow similar patterns. When the estimation error is small, the amount of regulation reserve used is small. However, these curves are not exactly the same, due to the wind curtailment when the realized wind generation is higher than the forecast.

Note that in the above simulations, the focus is on the case in which the forecast is higher than the realized wind generation, since this is the case that impacts ED. For the case that the forecast is lower than the realized wind generation, the forecast error would not impact the system much, since the over generated power from the wind farms will be curtailed and the system operator does not need to use regulation reserves to compensate for this forecast error.

5.4 Conclusion

In this chapter, joint optimization of ED and interruptible load management was investigated using short-term wind farm generation forecast. Specifically, a finite state Markov chain model for wind farm generation forecast was developed based on spatial and temporal characterizations of wind turbine power outputs. The proposed joint optimization of ED and interruptible load management was cast as a Markov decision process based dynamic programming problem. To reduce the complexity of this joint optimization problem, the greedy policy was used. Further, by leveraging the convexity properties, the proposed ED problem was transformed into a stochastic program by using the point forecast of wind generation. Numerical studies, via the IEEE Reliability Test System – 1996 and realistic wind measurement data from an actual wind farm, demonstrated the significant benefits obtained by integrating the Markov-chain-based forecast and the interruptible load management, compared with conventional wind-speed-based forecasting methods. In future work, it is of great interest to develop systematic approaches to address the uncertainty of wind generation in electricity markets in a cost-effective manner. For example, secondary ancillary services markets can be considered in the proposed ED framework.

Chapter 6

PRICING-BASED DECENTRALIZED SPECTRUM ACCESS CONTROL IN COGNITIVE RADIO NETWORKS

6.1 Introduction

Cognitive radio is expected to capture temporal and spatial "spectrum holes" in the spectrum white space, and to enable spectrum sharing for secondary users (SUs). One grand challenge is how SUs can discover spectrum holes and access them efficiently, without causing interference to the primary users (PUs), especially when the demand for available spectrum nearly outstrips the supply. Market-based mechanisms have been explored as a promising approach for spectrum access, where PUs can dynamically trade unused spectrum to SUs [6, 116–130]. In particular, auction-based spectrum access mechanisms have been extensively studied, including incentive compatibility [117–120], spectrum reuse [117, 118, 121, 122], auctioneer's revenue maximization [118], social welfare maximization [122], power allocation for the SUs with interference protection for the PU [123] (and the references therein). These works focus on on-demand auctions where each SU requests spectrum based on its traffic demand, and it is worth noting that the overhead can be significant in the auction procedure (e.g., market setup time, bidding time, and pricing clearing time).

Compared with auction-based spectrum access, pricing-based spectrum access incurs lower overhead (see [124–130] and the references therein). Notably, [124] studied pricing policies for a PU to sell unused spectrum to multiple SUs. Recent works [125,126] considered competition among multiple PUs that

sell spectrum, whereas [127] focused on competition among multiple SUs to access the PU's channels. [128] considered spectrum trading across multiple PUs and multiple SUs. [130] studied the investment and pricing decisions of a network operator under spectrum supply uncertainty. One common assumption used in these studies is that *orthogonal* multiple access is used among SUs, either in time or frequency domain, where a central controller is needed to handle SUs' admission control, to calculate the prices, and to charge the SUs. However, the computational complexity for dynamic spectrum access and the need of centralized controllers can often be overwhelming or even prohibitive. To address these problems, a recent work [129] proposed a two-tier market model based on the decentralized bargain theory, where the spectrum is traded from a PU to multiple SUs on a larger time scale, and then redistributed among SUs on a smaller time scale. Due to the decentralized nature, coordination among SUs remains a challenge when SUs of different networks coexist [6], simply because contention between SUs is unavoidable.

As a less-studied alternative in cognitive radio networks, random access can serve as a platform for the contention among SUs (e.g., in [131, 132]), and can be employed for decentralized spectrum access to mitigate the overwhelming complexity. With this insight, we focus on pricing-based spectrum control with random access. In particular, we study the behaviors of PUs and SUs in two spectrum trading market models based on random access: one with a monopoly PU market and the other with a multiple PU market.

We first consider the monopoly PU market model where the PU's unused spectrum is fixed. We study pricing-based dynamic spectrum access based on slotted Aloha, aiming to characterize the optimal pricing strategy maximizing the PU's revenue. Due to the nonconvexity of the optimization

problem, the global optimum is often unattainable. Instead, we first characterize the Pareto optimal region associated with the throughput vector of SUs, based on the observation that the global optimum has to be Pareto optimal. Roughly speaking, for any Pareto optimal solution, the throughput of any individual SU cannot be improved without deteriorating some other SU's throughput. Then, by maximizing the SUs' throughput subject to the budget constraints, we provide a Pareto optimal solution that is near-optimal. Furthermore, the structural properties of this Pareto optimal solution indicate that the access probabilities can be computed by the SUs locally. With this insight, we develop a decentralized pricing-based spectrum access control algorithm accordingly. To mitigate the spectrum underutilization due to the "price of contention," we next turn to dynamic spectrum access using CSMA, and quantify the improvements in spectrum utilization and PU's revenue. We also consider the case when PU's salable spectrum is controllable, i.e., the PU can flexibly allocate the spectrum to its ongoing transmissions so as to balance its own utility and revenue.

Next, for the multiple PU market model, we treat the competition among PUs as a three-stage Stackelberg game, where each PU seeks to maximize its net utility and each SU selects a PU's channel to maximize its own throughput. We explore optimal strategies to adapt the prices and the offered spectrum for each PU, and show that the Nash equilibria of the game exist. We further prove that the Nash equilibrium is unique when the number of PUs is less than a threshold, whose value is determined by the budgets and elasticity of SUs. Intuitively, this threshold criterion can be used by PUs to decide whether to join in the competition or not, i.e., when the number of PUs grows larger than the threshold, the competition among PUs is too strong, in-

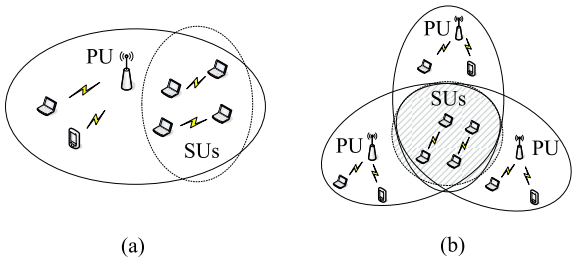

Figure 6.1: System model: (a) monopoly PU market (b) multiple PU market.

dicating that it is unprofitable for a PU to sell spectrum to SUs. An iterative algorithm is devised to compute the Nash equilibrium accordingly.

The rest of the chapter is organized as follows. In Section II, we study the monopoly PU market, and present the Pareto optimal pricing strategy for the PU's revenue maximization problem. We also characterize the tradeoff between the PU's utility and its revenue. We study in Section III the competition among PUs in the multiple PU market, which is cast as a three-stage Stackelberg game, and analyze the Nash equilibria of the game. Finally, we conclude the chapter in Section IV.

6.2 Monopoly PU Market

6.2.1 System Model

We first consider a monopoly PU market with a set of SUs, denoted by $\mathcal{M}$. The PU sells the available spectrum opportunity c in each period,[1] in terms of time slots in a slotted wireless system based on random access, to SUs who are willing to buy the spectrum opportunities (Fig. 6.1(a)). When one SU decides to buy the spectrum opportunity, it sends a request message together

[1]The period refers to a time frame where the PU sells its unused part, denoted by c. Please refer to Fig. 6.2 for an illustration of a period in the slotted wireless system.

Table 6.1: Summary of the key notation and definitions.

Notation	Definition
$\mathcal{M}$	set of SUs
$\mathcal{N}$	set of PUs
c	available spectrum opportunity
p_i (in vector $\mathbf{p}$)	usage price of SU i
g_i (in vector $\mathbf{g}$)	flat price of SU i
s_i (in vector $\mathbf{s}$)	successful channel access probability of SU i
z_i (in vector $\mathbf{z}$)	channel access probability of SU i
$U_i(\cdot)$	utility function of SU i
$d_i(\cdot)$	demand function of SU i
κ	spectrum utilization percentage
$R(\cdot)$	net utility function of PU
$\lvert\mathcal{N}\rvert$	cardinality of the set $\mathcal{N}$

with the budget information (to be elaborated in the sequel) to the PU. Then, at the beginning of each period, the PU broadcasts to the SUs the salable spectrum opportunities and the prices to access them. Observe that message passing involved in this scheme is infrequent and minimum (instead of sending the control message at each slot to manage the spectrum access).[2]

We study two cases: 1) the spectrum opportunity c is fixed and the PU desires to find the optimal prices (i.e., usage price and flat price) to maximize its revenue; 2) the spectrum opportunity c is a control parameter that the PU can use to balance its own utility and revenue. In both cases, each SU i seeks to set its demand that maximizes its payoff, given the spectrum opportunity c, the usage price p_i, and the flat price g_i. For ease of reference, the key notation in this chapter is listed in Table 6.1.[3]

[2]In cognitive radio networks, the control channel often does not exist [133]. In this study, assuming that no common control channel exist, we design algorithms that have minimal computational complexity and message passing overhead. The salable spectrum opportunity c is a fraction of a period. When selling the spectrum at the beginning of each period, the PU does not know which slots will be idle, as this depends on the PU's traffic, and SUs would have to detect the opportunities through spectrum sensing.

[3]We use bold symbols (e.g., $\mathbf{p}$) to denote vectors and calligraphic symbols (e.g., $\mathcal{M}$) to denote sets.

6.2.2 The Case with Fixed Spectrum Opportunity

We first study the case where the spectrum opportunity c is fixed. We begin with the channel access model for SUs, assuming a linear pricing strategy, i.e., the PU charges each SU a flat price and a usage price proportional to its *successful* transmissions. We will show that the Pareto optimal usage price is the same for all SUs, and is uniquely determined by the total demand of SUs and the available spectrum opportunity c.

1) Slotted Aloha Model for SUs' Channel Access

Each SU first carries out spectrum sensing to detect the PU's activity.[4] When the sensing result reveals that the PU is idle, SUs will contend for channel access by random access; otherwise, SUs will remain silent as illustrated in Fig. 6.2. As in the standard slotted Aloha model [134], we assume that SUs under consideration are within the contention ranges of the others and all transmissions are slot-synchronized. We assume that SUs always have packets to transmit and traffic demands of SUs are elastic. Denote by z_i the transmission probability of the ith SU. The probability that the ith SU's packet is successfully received is $s_i = z_i \prod_{j \neq i}(1 - z_j)$. The expected number of successfully transmitted packets of the ith SU in one period can be written as $\hat{s}_i = cs_i$, where $\hat{s}_i \in [0, c]$. Accordingly, the ith SU receives a utility in one period equal to $U_i(\hat{s}_i)$, where $U_i(\cdot)$ denotes the utility function of the ith SU.

[4] In this study, we assume that sensing errors are negligible.

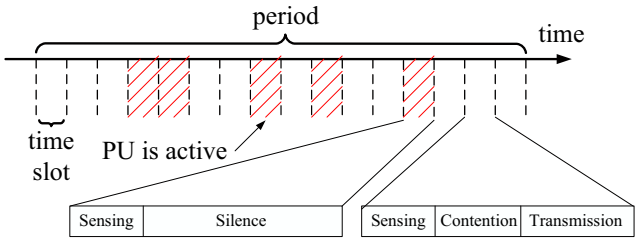

Figure 6.2: Random access model for SUs.

The optimal demand $\hat{s}_i^*$ is the solution to the following optimization problem:

$$\begin{aligned} &\text{maximize} && U_i(\hat{s}_i) - (p_i \hat{s}_i + g_i) \\ &\text{subject to} && 0 \leq \hat{s}_i \leq c, \\ &\text{variables} && \{\hat{s}_i\}. \end{aligned} \tag{6.1}$$

As is standard, we define the demand function that captures the successful transmissions $\hat{s}_i$ given the price p_i as

$$d_i(p_i) = \begin{cases} U_i'^{-1}(p_i) & \text{if } g_i \leq U_i(\hat{s}_i) - p_i \hat{s}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

Assuming α-fair utility functions, the utility and the demand functions of each SU can be written, respectively, as

$$U_i\left(\hat{s}_i\right) = \begin{cases} \frac{\sigma_i}{1-\alpha}\left(\hat{s}_i\right)^{1-\alpha} & 0 \leq \alpha < 1, \\ \sigma_i \log\left(\hat{s}_i\right) & \alpha = 1, \end{cases} \tag{6.3}$$

$$d_i(p_i) = \begin{cases} \sigma_i^{\frac{1}{\alpha}} p_i^{-\frac{1}{\alpha}} & \text{if } g_i \leq U_i(\hat{s}_i) - p_i \hat{s}_i, \\ 0 & \text{otherwise,} \end{cases} \tag{6.4}$$

where σ_i, the multiplicative constant in the α-fair utility function, denotes the utility level[5] of the ith SU, which reflects the *budget*[6] of the ith SU (see [135]

[5]The PU collects the budget information of SUs from the SUs' request message at the beginning of each period, and this allows the PU to infer σ_i.

[6]Given the prices, the SUs with larger budget, i.e., larger σ_i, would buy more spectrum from the PU based on (6.4).

and the references therein). Note that we mainly consider the case of $0 \leq \alpha \leq 1$ because $\frac{1}{\alpha} = -\frac{p_i d_i'(p_i)}{d_i(p_i)}$ is the elasticity of the demand seen by the PU, and it has to be strictly larger than 1 so that the monopoly price is finite [136]. Clearly, the α-fairness boils down to the weighted proportional fairness when $\alpha = 1$, and to selecting the SU with the highest budget when $\alpha = 0$.

2) PU's Pricing Strategy

We have the following revenue maximization problem for the monopoly PU market:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i\in\mathcal{M}} (g_i + p_i \hat{s}_i) \\ \text{subject to} \quad & \hat{s}_i \leq c z_i \textstyle\prod_{k\neq i}(1 - z_k),\ \forall i \in \mathcal{M}, \\ & \hat{s}_i = d_i(p_i),\ \forall i \in \mathcal{M}, \\ & U_i(\hat{s}_i) - g_i - p_i \hat{s}_i \geq 0,\ \forall i \in \mathcal{M}, \\ \text{variables} \quad & \{\mathbf{g}, \mathbf{p}, \hat{\mathbf{s}}, \mathbf{z}\}. \end{aligned} \tag{6.5}$$

The constraint $U_i(\hat{s}_i) - g_i - p_i \hat{s}_i \geq 0$ ensures that SUs have non-negative utility under the prices g_i and p_i; otherwise, SUs may not transmit. Let $\{g_i^*, p_i^*, \hat{s}_i^*, z_i^*\}$ denote the optimal solution to (6.5). It is clear that $U_i(\hat{s}_i^*) - g_i^* - p_i^* \hat{s}_i^* = 0,\ \forall i \in \mathcal{M}$; otherwise, the PU can always increase its revenue by increasing g_i^* to make the SU's net utility equal to zero.

Lemma 6.2.1. *The optimal prices for (6.5) are given by*

$$g_i = U_i(\hat{s}_i) - p_i \hat{s}_i,\ \forall i \in \mathcal{M}. \tag{6.6}$$

Based on Lemma 6.2.1, (6.5) can be rewritten as:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i\in\mathcal{M}} U_i(d_i(p_i)) \\ \text{subject to} \quad & d_i(p_i) \leq c z_i \textstyle\prod_{k\neq i}(1 - z_k),\ \forall i \in \mathcal{M}, \\ \text{variables} \quad & \{\mathbf{p}, \mathbf{z}\}. \end{aligned} \tag{6.7}$$

Since the utility function $U_i(\cdot)$ is increasing, the optimal solution to (6.7) is achieved at the point when $d_i(p_i) = cz_i \prod_{k \neq i}(1 - z_k),\ \forall i \in \mathcal{M}$. Also, the objective function of (6.7) can be written as $U_i(d_i(p_i)) = \frac{1}{1-\alpha} p_i d_i(p_i)$. Since $\frac{1}{1-\alpha}$ is a constant, maximizing $\frac{1}{1-\alpha} \sum_{i \in \mathcal{M}} p_i d_i(p_i)$ is equivalent to maximizing $\sum_{i \in \mathcal{M}} p_i d_i(p_i)$, i.e., solving (6.5) is equivalent to solving the following problem without considering flat prices:

$$\begin{array}{ll} \text{maximize} & \sum_{i \in \mathcal{M}} p_i d_i(p_i) \\ \text{subject to} & d_i(p_i) = cz_i \prod_{k \neq i}(1 - z_k),\ \forall i \in \mathcal{M}, \\ \text{variables} & \{\mathbf{p}, \mathbf{z}\}. \end{array} \tag{6.8}$$

In general, (6.8) is nonconvex, and therefore it is difficult to find the global optimum. Observing that the global optimum of (6.8) is Pareto optimal, we shall confine our attention to the Pareto optimal region, i.e., the set consisting of Pareto optimal solutions to (6.8).

Definition 6.2.1. *A feasible allocation $\hat{\mathbf{s}}$ is Pareto optimal if there is no other feasible allocation $\hat{\mathbf{s}}'$ such that $\hat{s}'_i \geq \hat{s}_i$ for all $i \in \mathcal{M}$ and $\hat{s}'_j > \hat{s}_j$ for some $j \in \mathcal{M}$.*

Lemma 6.2.2. *The Pareto optimal region corresponding to (6.8) has the following properties:*

1. *The global optimum is in the Pareto optimal region.*
2. *The solution to (6.8) is Pareto optimal if and only if $\sum_{i \in \mathcal{M}} z_i = 1$.*

The proof of Lemma 6.2.2 is given in Appendix 6.5.1. By Lemma 6.2.2, for any Pareto optimal allocation $d_i(p_i),\ \forall i \in \mathcal{M}$, we have $\sum_{i \in \mathcal{M}} z_i = 1$. Let $\mathcal{A} = \{\mathbf{z} | \sum_{i \in \mathcal{M}} z_i = 1, z_i \geq 0,\ \forall i \in \mathcal{M}\}$ be the Pareto optimal region. Therefore, it suffices to search for the points in $\mathcal{A}$ which can maximize (6.8).

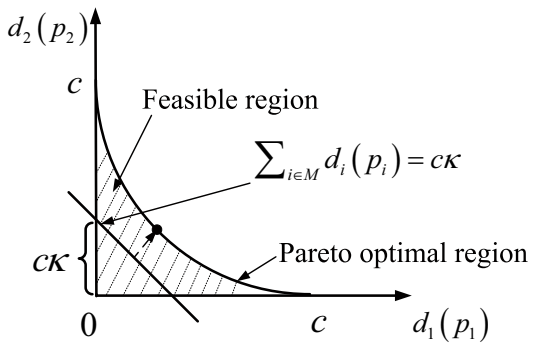

Figure 6.3: A sketch of the Pareto optimal region: the case with two SUs.

In light of Lemma 6.2.2, instead of tackling the original problem given by (6.8), hereafter we focus on obtaining a Pareto optimal solution to (6.8) that maximizes the sum of SUs' throughput given the SUs' budget constraints, by confining the search space to the hyperplane $\sum_{i\in\mathcal{M}} d_i(p_i) = c\kappa$, where $\kappa \in [0,1]$ denotes the *spectrum utilization percentage* under the allocation $d_i(p_i),\ \forall i \in \mathcal{M}$.

We now consider this "constrained" version of (6.8), for finding the maximum feasible spectrum utilization κ^*, i.e., the tangent point of the hyperplane and $\mathcal{A}$, as illustrated in Fig. 6.3:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i\in\mathcal{M}} p_i d_i(p_i) \\ \text{subject to} \quad & \textstyle\sum_{i\in\mathcal{M}} d_i(p_i) = c\kappa, \\ & d_i(p_i) = c z_i \textstyle\prod_{k\neq i}(1-z_k),\ \forall i \in \mathcal{M}, \\ \text{variables} \quad & \{\mathbf{p}, \mathbf{z}, \kappa\}. \end{aligned} \tag{6.9}$$

We note that the solution to (6.9) is in general suboptimal for (6.8). However, by exploring the connections between the pricing strategy and the spectrum utilization, we derive a closed form solution to (6.9) that is also a near-optimal solution to (6.8).

Proposition 6.2.1. *For $\alpha \in (0,1)$, the optimal solution to (6.9) is given by*

$$\begin{array}{rcl} p_i^* & = & \left(\frac{G}{c\kappa^*}\right)^{\alpha},\ \forall i \in \mathcal{M}, \\ g_i^* & = & U_i(d_i(p_i^*)) - p_i^* d_i(p_i^*),\ \forall i \in \mathcal{M}, \\ \kappa^* & = & \sum_{i\in\mathcal{M}} z_i^* \prod_{j\neq i}(1-z_j^*), \\ z_i^* & = & \frac{w_i}{w_i+e^{-u}},\ \forall i \in \mathcal{M}, \end{array} \tag{6.10}$$

where $w_i = \sigma_i^{\frac{1}{\alpha}} G^{-1}$, $G = \sum_{i\in\mathcal{M}} \sigma_i^{\frac{1}{\alpha}}$, and u is the unique solution of

$$\sum_{i\in\mathcal{M}} \frac{w_i}{w_i+e^{-u}} = 1.$$

Proof outline: In what follows, we sketch a outline for proving the above proposition (see Appendices 6.5.2 and 6.5.3 for the proof details). First, the following result establishes the relationship between the optimal pricing strategy of (6.9) and κ when $\alpha \in (0,1)$.

Lemma 6.2.3. *Given κ, the optimal pricing strategy of (6.9) for $\alpha \in (0,1)$ is given by*

$$p_i^* = \left(\frac{G}{c\kappa}\right)^{\alpha},\ \forall i \in \mathcal{M}. \tag{6.11}$$

Next, a key step is to find κ^*. By Lemma 6.2.3, we have $d_i(p_i^*) = \sigma_i^{\frac{1}{\alpha}} G^{-1} c\kappa,\ \forall i \in \mathcal{M}$. Utilizing those constraints, we can find κ^* by solving the following problem:

$$\begin{array}{ll} \text{maximize} & \kappa \\ \text{subject to} & w_i\kappa \leq z_i \prod_{j\neq i}(1-z_j),\ \forall i \in \mathcal{M}, \\ \text{variables} & \{\kappa, \mathbf{z}\}. \end{array} \tag{6.12}$$

Still, (6.12) is nonconvex, but by first taking logarithms of both the objective function and the constraints and then letting $\kappa' = \log(\kappa)$, we can

transform (6.12) into the following convex problem:

$$\begin{array}{ll} \text{maximize} & \kappa' \\ \text{subject to} & \forall i \in \mathcal{M} \\ & \log(w_i) + \kappa' \leq \log(z_i) + \sum_{j \neq i} \log(1 - z_j), \\ \text{variables} & \{\kappa', \mathbf{z}\}. \end{array} \tag{6.13}$$

Thus, the optimal solution to (6.12) can be summarized by the following lemma.

Lemma 6.2.4. *The optimal solution to (6.12) is given by*

$$\begin{array}{rcl} \kappa^* & = & \sum_{i \in \mathcal{M}} z_i^* \prod_{j \neq i} (1 - z_j^*), \\ z_i^* & = & \frac{w_i}{w_i + e^{-u}}, \ \forall i \in \mathcal{M}, \end{array} \tag{6.14}$$

where u is the unique solution of

$$\sum_{i \in \mathcal{M}} \frac{w_i}{w_i + e^{-u}} = 1. \tag{6.15}$$

Further, when the number of SUs in the network is large, i.e., $|\mathcal{M}| \to \infty$, we can approximate κ^* by e^{-1}.

Based on Lemma 6.2.3 and Lemma 6.2.4, the optimal solution to (6.9) is given by Proposition 6.2.1, which is a Pareto optimal solution to (6.5). □

Remarks: The "constrained" revenue maximization problem (6.9) is nonconvex. However, (6.9) exhibits a *hidden convexity* property, i.e., by utilizing Lemma 6.2.3, (6.9) can be transformed into a convex problem (6.13). Intuitively speaking, this property ensures that the line, $\sum_{i \in \mathcal{M}} d_i(p_i) = c\kappa$, as illustrated in Fig. 6.3, can touch the Pareto optimal region with one unique point.

So far we have focused on the case with $\alpha \in (0, 1)$. Next we consider the special cases when $\alpha = 0$ and 1. Interestingly, we will see that the Pareto optimal solutions are also global optimal in those special cases.

Corollary 6.2.1. *When $\alpha = 0$, the Pareto optimal solution to (6.5) is also the global optimal solution, which is given by*

$$\begin{aligned} p_i^* &= \max_{i\in\mathcal{M}} \sigma_i, \\ z_i^* &= \begin{cases} 1, \ i = k \\ 0, \ i \neq k \end{cases}, \\ g_i^* &= 0, \end{aligned} \tag{6.16}$$

for all i, where $k = \arg\max_{j\in\mathcal{M}} \sigma_j$.

Corollary 6.2.1 implies that, when $\alpha = 0$, the PU selects the SU with the highest budget and only allows that SU to access the channel with probability 1 to maximize its revenue.

Remarks: The Pareto optimal solution in (6.10) converges to the globally optimal solution as α goes to zero, i.e.,

$$p_i^* = \left(\tfrac{G}{c\kappa^*}\right)^\alpha = \left(\tfrac{1}{c\kappa^*}\right)^\alpha \left(\sum_{j\in\mathcal{M}} \sigma_j^{\frac{1}{\alpha}}\right)^\alpha \to \max_{j\in\mathcal{M}} \sigma_j.$$

When $\alpha = 1$, (6.7) can be transformed into a convex problem by taking logarithms of the constraints, and the global optimal access probabilities of SUs in this case are

$$z_i^* = \frac{\sigma_i}{\sum_{k\in\mathcal{M}} \sigma_k}, \ \forall i \in \mathcal{M},$$

i.e., the random access probability is proportional to SU's utility level, where the revenue is dominated by the flat rate. This is similar to the observation made in [135]. As α approaches 1, the revenue computed by (6.10) also converges to the global optimal solution, since the Pareto optimal flat rates in (6.10) converge to the global optimal ones.

3) Decentralized Implementation

Based on the above study, we next develop decentralized implementation of the pricing-based spectrum access control. Based on the structural properties of the Pareto optimal solution given in Proposition 6.2.1, we develop a decentralized pricing-based spectrum access control algorithm (Algorithm 5). In particular, the PU only needs to compute and broadcast the common parameters p^*, G and u, based on which each SU can compute its access probability locally. It is clear that Algorithm 5 significantly reduces the complexity and the amount of the message passing, which would otherwise require a centralized coordination for the PU.

Algorithm 5 Decentralized Pricing-based Spectrum Access Control

Initialization:
1) The PU collects the budget information of SUs, i.e., $\{\sigma_i\}$.
2) The PU computes p^*, $\mathbf{g}^*$, G and u by (6.10), and broadcasts p^*, G and u to SUs.
3) Each SU computes z_i^* by (6.10) based on G and u, and infers $\mathbf{g}^*$ from its own utility and p^* by (6.10).
Repeat at the beginning of each period:
If New SUs join the system **or** SUs leave the system **then**
The PU updates the budget information of SUs. Then run Steps 2 to 3.
Endif

6.2.3 Numerical Example: Pareto Optimum vs. Global Optimum

To reduce the computational complexity in solving the global optimum of (6.5), we first solve (6.9). To examine the efficiency of this Pareto optimal solution, we exhaustively search for the global optimum of (6.5) to compare with the Pareto optimum, in a small network with three SUs so as to efficiently generate the true global optimum as the benchmark. In this example, c is set to 5, and each SU's σ_i is generated uniformly in the interval $[0, 4]$ and fixed for different α for the sake of comparison. As shown in Fig. 6.4, the Pareto optimal solution is close to the global optimal solution. Furthermore, the gap between the objective value evaluated at the Pareto optimal solution and the

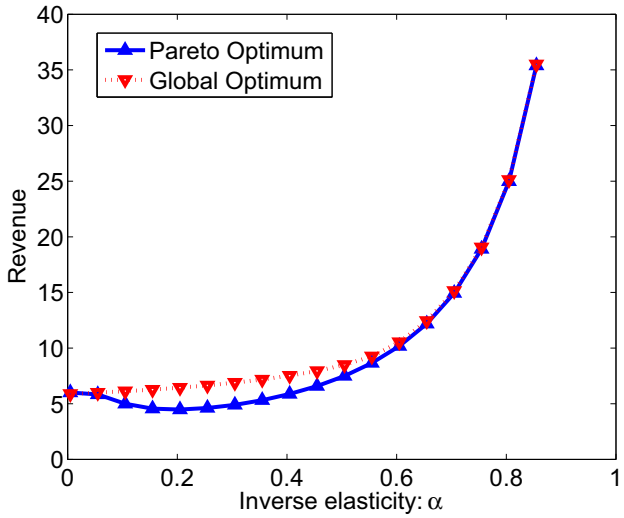

Figure 6.4: Pareto optimum vs. global optimum.

global optimal value diminishes as α approaches 1. In addition, the gap goes to zero as α goes to zero, corroborating Corollary 6.2.1.

6.2.4 CSMA Model for SUs' Channel Access

Needless to say, the contention among SUs leads to spectrum under-utilization. For the slotted Aloha model, the spectrum utilization κ approaches $\frac{1}{e}$, when the network size grows large, indicating that the unused spectrum is $1 - \frac{1}{e}$.

Definition 6.2.2. *We define the unused spectrum, $1 - \kappa$, as the "price of contention".*

Obviously, the "price of contention" using slotted Aloha is high, compared to orthogonal access that requires centralized control. It is well known that spectrum utilization can be enhanced by using CSMA. Thus motivated, next we consider a CSMA-based random access for the SUs' channel access.

When CSMA is employed, a SU can successfully access the channel

after an idle period if no other SUs attempt to access the channel at the same time. Let β denote the idle time of the channel. For a given $\mathbf{z}$, the network service throughput for a large network can be approximated by [134]:

$$C(T(\mathbf{z})) = \frac{T(\mathbf{z})e^{-T(\mathbf{z})}}{\beta+1-e^{-T(\mathbf{z})}}, \tag{6.17}$$

and the successful channel access probability of the ith SU can be approximated by

$$s_i(\mathbf{z}) = \frac{z_i e^{-T(\mathbf{z})}}{\beta+1-e^{-T(\mathbf{z})}},\ \forall i \in \mathcal{M}, \tag{6.18}$$

where $T(\mathbf{z}) = \sum_{i\in\mathcal{M}} z_i$ denotes the rate at which the SUs attempt to access the channel at the end of an idle period.

Since $C(T(\mathbf{z}))$ is maximized when $T(\mathbf{z}) = \sqrt{2\beta}$, the PU can always adjust the prices to make the SUs' channel access rates satisfy $\sum_{i\in\mathcal{M}} z_i = \sqrt{2\beta}$. In this case, $\frac{c}{1+\sqrt{2\beta}}$ can be utilized by the SUs per period, i.e., the spectrum utilization under CSMA is $\frac{1}{1+\sqrt{2\beta}}$. Similar to the approach for solving (6.5), by confining the search space to $\sum_{i\in\mathcal{M}} z_i = \sqrt{2\beta}$, the "constrained" revenue maximization problem under CSMA is given by:[7]

$$\begin{array}{ll} \text{maximize} & \sum_{i\in\mathcal{M}} (g_i + p_i\hat{s}_i) \\ \text{subject to} & \hat{s}_i \le \frac{cz_i e^{-\sqrt{2\beta}}}{\beta+1-e^{-\sqrt{2\beta}}},\ \forall i \in \mathcal{M}, \\ & \hat{s}_i = d_i(p_i),\ \forall i \in \mathcal{M}, \\ & \sum_{i\in\mathcal{M}} z_i = \sqrt{2\beta}, \\ & U_i(\hat{s}_i) - g_i - p_i\hat{s}_i \ge 0,\ \forall i \in \mathcal{M}, \\ \text{variables} & \{\mathbf{g}, \mathbf{p}, \hat{\mathbf{s}}, \mathbf{z}\}. \end{array} \tag{6.19}$$

The above problem can be solved by following the same approach to (6.5), and the optimal solution to (6.19) is given by the following proposition.

[7] The solution to (6.19) is a Pareto optimal solution to the original revenue maximization problem under CSMA without the constraint $\sum_{i\in\mathcal{M}} z_i = \sqrt{2\beta}$.

Proposition 6.2.2. *For $\alpha \in (0,1)$, the optimal solution to (6.19) is given by*

$$\begin{aligned} p_i^* &= \left(\tfrac{(1+\sqrt{2\beta})G}{c}\right)^{\alpha}, \ \forall i \in \mathcal{M}, \\ g_i^* &= U_i(d_i(p_i^*)) - p_i^* d_i(p_i^*), \ \forall i \in \mathcal{M}, \\ z_i^* &= \tfrac{\sigma_i^{\frac{1}{\alpha}}\sqrt{2\beta}}{G}, \ \forall i \in \mathcal{M}. \end{aligned} \tag{6.20}$$

Remarks: Note that (6.17) and (6.18) offer good approximations for large networks, in which case we can compare the performance of slotted Aloha and CSMA, e.g., revenue and spectrum utilization. As expected, the spectrum utilization under CSMA is higher, resulting in higher revenue from the SUs. We caution that, when the network size is small, such comparisons may not be accurate since the system capacity under CSMA is unknown. Under slotted Aloha, the results hold for an arbitrary number of SUs. In the remaining of the chapter, we focus on the system under slotted Aloha only.

6.2.5 The Case with Controllable Spectrum Opportunity

When the spectrum opportunity c is a control parameter, there exists a tradeoff between the PU's utility and its revenue. For ease of exposition, we use the logarithmic utility function to quantify the PU's satisfaction:

$$V(c) = a \log\left(1 - \tfrac{c}{C}\right), \tag{6.21}$$

where C denotes the total length of a period, and the utility level a is a positive constant depending on the application type [128]. Given the PU's utility function, the net utility (or profit) of the PU for $\alpha \in (0,1)$ can be expressed as:

$$\begin{aligned} R(c,\mathbf{p}) &= a \log\left(1 - \tfrac{c}{C}\right) + \textstyle\sum_{i\in\mathcal{M}} \left(g_i + p_i d_i(p_i)\right) \\ &= a \log\left(1 - \tfrac{c}{C}\right) + \tfrac{1}{1-\alpha} \textstyle\sum_{i\in\mathcal{M}} p_i d_i(p_i). \end{aligned} \tag{6.22}$$

Based on Proposition 6.2.1, the Pareto optimal price is a function of the spectrum opportunity c. Then, the optimal c^* can be found by solving the following

problem:

$$\begin{aligned} &\text{maximize} && a\log\left(1-\tfrac{c}{C}\right)+bc^{1-\alpha} \\ &\text{subject to} && c\in\{0,1,\ldots,C\}, \\ &\text{variable} && \{c\}, \end{aligned} \tag{6.23}$$

where $b=\frac{1}{1-\alpha}\kappa^{*1-\alpha}G^{\alpha}$ is a positive constant.

Note that c is constrained to be an integer in the slotted system noted above. However, the objective function of (6.23) has the unimodal property and therefore (6.23) can be solved efficiently (e.g., by the Fibonacci search algorithm [47]).

Lemma 6.2.5. *The objective function of (6.23) is unimodal for* $c\in\{0,1,\ldots,C\}$.

The unimodal property of the objective function of (6.23) directly follows from that of its continuous version, whose optimal solution c^* can be determined by the first order condition and the boundary conditions, which is the solution to

$$-\tfrac{a}{C-c^*}+\tfrac{b}{1-\alpha}c^{*-\alpha}=0. \tag{6.24}$$

When the length of each period, C, is reasonably large, c^* can be approximated by the solution to (6.24). In what follows, we adopt this continuous approximation of c.

As an illustration, we plot two possible curves of (6.22) for different b in Fig. 6.5. In this example, we set $a=20$, $C=100$, $\alpha=0.5$ and $|\mathcal{M}|=20$. Each SU's σ_i is generated uniformly in the interval $[\sigma_{\min},\sigma_{\max}]$. For the two realizations of $\sigma_i,\ \forall i\in\mathcal{M}$, the optimal tradeoff decision, corresponding to the highest point of each curve within $(0,C)$, increases with b. Intuitively speaking, the PU would allocate more spectrum opportunity to those SUs who would pay more.

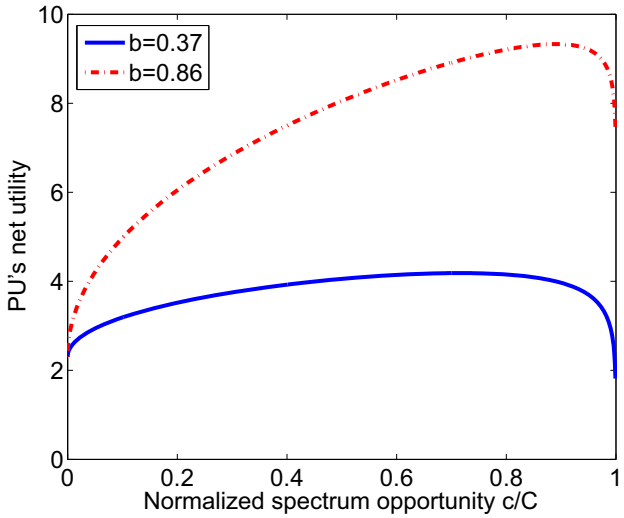

Figure 6.5: PU's profit under different b.

6.3 Multiple PU Market

6.3.1 System Model

When there are multiple PUs in a cognitive radio network, they compete with each other in terms of prices and spectrum opportunities, in order to maximize their net utilities. It can be seen from Lemma 6.2.1 that PU's flat prices depend on usage prices and that each SU wishes to choose the PU with the lowest usage price for its transmission.[8] Thus, we focus on usage prices in what follows, and the corresponding flat prices can be obtained accordingly.

We assume that both PUs and SUs are selfish and yet rational. As the PUs are spectrum providers, they have the right to decide the prices and spectrum opportunities, so as to maximize their net utilities. Based on PU's decisions, each SU then chooses a PU's channel to maximize its transmission

[8]Based on Lemma 6.2.1, each SU ends up with zero profit in any PU's channel. However, the SU can maximize its transmission rate by choosing the PU with the lowest usage price. Furthermore, the access probability of each SU is determined by (6.10).

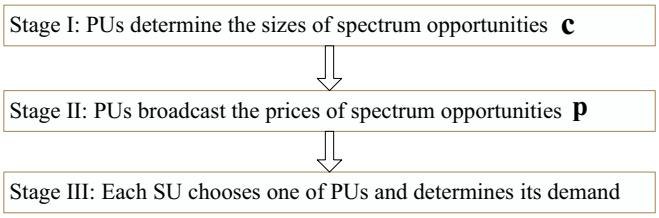

Figure 6.6: A three-stage Stackelberg game.

rate. Observe that it is a typical leader-follower game which can be analyzed by using the Stackelberg game framework. Specifically, we cast the competition among the PUs as a three-stage Stackelberg game, as summarized in Fig. 6.6, where the PUs and the SUs adapt their decisions dynamically to reach an equilibrium point. The PUs first simultaneously determine in Stage I their available spectrum opportunities, and then in Stage II simultaneously announce the prices to the SUs. Finally, each SU accesses only one PU's channel to maximize its throughput in Stage III. Here, we consider a set $\mathcal{N}$ of PUs. We assume that all SUs are within the intersection of those PU's coverage areas shown in Fig. 6.1(b).

In the sequel, we focus on the game for $\alpha \in (0, 1)$, and use the index $i \in \mathcal{M}$ for SUs and the index $j \in \mathcal{N}$ for PUs.

6.3.2 Backward Induction for the Three-stage Game

We analyze the subgame perfect equilibrium of the game by using the backward induction method [136], which is a popular technique for determining the subgame perfect equilibrium. First, we start with Stage III and analyze SU's behaviors, under given PU's spectrum opportunities and prices. Then we turn our focus to Stage II and analyze how PUs determine prices given spectrum opportunities and the possible reactions of SUs in Stage III. Finally, we

study how PUs determine spectrum opportunities given the possible reactions in Stage II and III.

1) Channel Selection in Stage III

In this stage, each SU determines which PU's channel to access based on the set of prices $\mathbf{p}$. The admission of SUs also depends on the available spectrum opportunities $\mathbf{c}$ in Stage I. Since (6.4) decreases with price p_j, the ith SU would choose the jth PU's channel if $p_j = \min_{k\in\mathcal{N}} p_k$.

Given the set of prices $\mathbf{p}$, the total demand of SUs in the jth PU's channel can be written as

$$D_j(p_j, \mathbf{p_{-j}}) = \sum_{i\in\mathcal{M}_j} d_i(p_j) = \sum_{i\in\mathcal{M}_j} \sigma_i^{\frac{1}{\alpha}} p_j^{-\frac{1}{\alpha}},$$

where $\mathcal{M}_j$ denotes the set of SUs choosing the jth PU, and $\mathbf{p_{-j}}$ denotes the set of prices of PUs other than the jth PU. Both $\mathcal{M}_j$ and D_j depend on prices $\mathbf{p}$, and are independent of $\mathbf{c}$. Therefore, the demand function can be written as

$$D_j(p_j, \mathbf{p_{-j}}) = \begin{cases} \frac{G}{|\mathcal{J}|} p_j^{-\frac{1}{\alpha}}, & j \in \mathcal{J} \\ 0, & j \notin \mathcal{J} \end{cases},$$

where G is defined in Proposition 6.2.1, and $\mathcal{J} = \{j | p_j = \min_{k\in\mathcal{N}} p_k, j \in \mathcal{N}\}$ denotes the set of PUs with the smallest price in $\mathcal{N}$. In this chapter, we assume that the SUs randomly pick one PU in $\mathcal{J}$ with equal probability.

Given the size of available spectrum opportunities c_j, the jth PU always adjusts its price to make the demand of SUs equal to the supply so as to maximize its revenue (based on Proposition 6.2.1). It follows that at the Nash equilibrium point,

$$D_j(p_j, \mathbf{p_{-j}}) = c_j \kappa_j^*, \quad \forall j \in \mathcal{N}, \tag{6.25}$$

where κ_j^* denotes the maximum feasible spectrum utilization of the jth PU's channel, and is given by Lemma 6.2.4. Since κ_j^* depends on the budgets of SUs, it is difficult for each PU to decide whether or not to admit new SUs based on the current demand. Note that, in the multiple PU market, the available spectrum is much larger than the single PU case, which can accommodate a large number of SUs. Since κ_j^* approaches e^{-1} when the number of SUs is reasonably large based on Lemma 6.2.4, we will approximate using this asymptote. Accordingly, (6.25) can be rewritten as

$$D_j(p_j, \mathbf{p_{-j}}) = \frac{c_j}{e}, \quad \forall j \in \mathcal{N}. \tag{6.26}$$

2) Pricing Competition in Stage II

In this stage, the PUs determine their pricing strategies while considering the demands of SUs in Stage III, given the available spectrum opportunities $\mathbf{c}$ in Stage I. The profit of the jth PU can be expressed as

$$R_j(c_j, p_j, \mathbf{c_{-j}}, \mathbf{p_{-j}}) = a_j \log\left(1 - \frac{c_j}{C}\right) + \frac{1}{1-\alpha} p_j D_j(p_j, \mathbf{p_{-j}}).$$

Since c_j is fixed at this stage, the jth PU is only interested in maximizing the revenue $p_j D_j(p_j, \mathbf{p_{-j}})$. Obviously, if the jth PU has no available spectrum to sell, i.e., $c_j = 0$, it would not compete with other PUs by price reduction to attract the SUs. For convenience, define $\mathcal{N}' = \{j | c_j > 0,\ \forall j \in \mathcal{N}\}$ as the set of PUs with positive spectrum opportunity.

Game at Stage II: The competition among PUs in this stage can be modeled as the following game:

- *Players*: The PUs in the set $\mathcal{N}'$;
- *Strategy*: Each PU can choose a price p_j from the feasible set $\mathcal{P} = [p_{\min}, \infty)$;

- *Objective function*: $p_j D_j(p_j, \mathbf{p_{-j}}), j \in \mathcal{N}'$,

where $p_{\min}$ denotes the minimum price that each PU can choose, and is determined by (6.10) at $c = C$.

Proposition 6.3.1. *A necessary and sufficient condition for PUs to achieve a Nash equilibrium price is* $\sum_{j\in\mathcal{N}'} c_j \leq C$. *Moreover, when* $\sum_{j\in\mathcal{N}'} c_j \leq C$, *there exists a unique Nash equilibrium price, and the Nash equilibrium price is given by* $p_j^* = p^*, \forall j \in \mathcal{N}'$, *where*

$$p^* = (\tfrac{eG}{\sum_{j\in\mathcal{N}'} c_j})^{\alpha}. \tag{6.27}$$

The proof of Proposition 6.3.1 is given in Appendix 6.5.4. Proposition 6.3.1 shows that no PU would announce a price higher than its competitors to avoid losing most or all of its demand to its competitors, and the optimal strategy is to make the same decision as its competitors. Since $c_j = 0, \forall j \notin \mathcal{N}'$, the equilibrium price (6.27) can be rewritten as

$$p^* = (\tfrac{eG}{\sum_{j\in\mathcal{N}} c_j})^{\alpha}, \tag{6.28}$$

where $\sum_{j\in\mathcal{N}} c_j \leq C$.

3) Spectrum Opportunity Allocation in Stage I

In this stage, the PUs need to decide the optimal spectrum opportunities to maximize their profits. Based on Proposition 6.3.1, the jth PU's profit can be written as

$$\begin{aligned} R_j(c_j, \mathbf{c_{-j}}) &= a_j \log\left(1 - \tfrac{c_j}{C}\right) + \tfrac{c_j}{e(1-\alpha)} \left(\tfrac{eG}{\sum_{k\in\mathcal{N}} c_k}\right)^{\alpha} \\ &= \tfrac{e^{\alpha-1}G^{\alpha}}{1-\alpha} \left(\hat{a}_j \log\left(1 - \tfrac{c_j}{C}\right) + \tfrac{c_j}{(\sum_{k\in\mathcal{N}} c_k)^{\alpha}}\right), \end{aligned}$$

where $\hat{a}_j = (1-\alpha)e^{1-\alpha}G^{-\alpha}a_j$. For convenience, define $\hat{R}_j(c_j, \mathbf{c_{-j}}) = \hat{a}_j \log\left(1 - \frac{c_j}{C}\right) + \frac{c_j}{(\sum_{k\in\mathcal{N}} c_k)^\alpha}$. Since $\frac{e^{\alpha-1}G^\alpha}{1-\alpha}$ is a constant, maximizing $R_j(c_j, \mathbf{c_{-j}})$ is equivalent to maximizing $\hat{R}_j(c_j, \mathbf{c_{-j}})$.

Game at Stage I: The competition among the PUs in this stage can be modeled as the following game:

- *Players*: The PUs in the set $\mathcal{N}$;
- *Strategy*: The PUs will choose $\mathbf{c}$ from the feasible set $\mathcal{C} = \{\mathbf{c} | \sum_{j\in\mathcal{N}} c_j \leq C, c_j \in [0, C),\ \forall j \in \mathcal{N}\}$;
- *Objective function*: $\hat{R}_j(c_j, \mathbf{c_{-j}})$, $\forall j \in \mathcal{N}$.

We first examine the existence of the Nash equilibrium of the game at this stage. Based on [137], the existence of the Nash equilibrium can be obtained by showing the concavity of $\hat{R}_j(c_j, \mathbf{c_{-j}})$ in terms of c_j.

Proposition 6.3.2. *There exists a Nash equilibrium in the spectrum opportunity allocation game, which satisfies the following set of equations:*

$$\hat{R}_j(c_j^*, \mathbf{c}_{-\mathbf{j}}^*) = \hat{a}_j \log\left(1 - \frac{c_j^*}{C}\right) + \frac{c_j^*}{(\sum_{k\in\mathcal{N}} c_k^*)^\alpha}, \forall j \in \mathcal{N}. \tag{6.29}$$

In general, the Nash equilibrium that satisfies (6.29) is not necessarily unique, as illustrated by the following example. Suppose a market with two heterogeneous PUs, with $\hat{a}_1 = 2.5$ and $\hat{a}_2 = 3.8$. Let $C = 20$ and $\alpha = 0.3$. Then, $\mathbf{c}^* = (12.45, 7.55)$ and $\mathbf{c}^* = (11.36, 8.64)$ are two possible Nash equilibria that satisfy (6.29).

In what follows, we provide a necessary and sufficient condition for the uniqueness of Nash equilibrium in the market with homogeneous PUs (i.e., $\hat{a}_j = \hat{a},\ \forall j \in \mathcal{N}$).

To find the Nash equilibrium of the game at this stage, we first examine the strategy of the jth PU given other PU's decisions. By checking the first order condition $\partial \hat{R}_j(c_j, \mathbf{c_{-j}})/\partial c_j = 0$ and the boundary conditions, we can obtain the best response strategy of the jth PU. As expected, the best response strategy for the jth PU depends on $\hat{a}$ and its competitors' decision $\mathbf{1^T c_{-j}}$. Let c^L and c^H be the thresholds for PU's decision making associated with $0 < \hat{a} \leq C^{1-\alpha}$ and $\hat{a} > C^{1-\alpha}$ respectively. They are given explicitly by $c^L = \frac{1}{2\alpha}(\alpha - 1 + \sqrt{(\alpha-1)^2 + 4\alpha\hat{a}C^{\alpha-1}})$ and $c^H = \left(\frac{\hat{a}}{C}\right)^{-\frac{1}{\alpha}}$ (derivation in Appendix 6.5.5). We now establish the response strategy for the PUs.

Proposition 6.3.3. *The best response strategy for the jth PU in the above game is outlined as follows:*

1. *The case with $0 < \hat{a} \leq C^{1-\alpha}$:*

   If $\mathbf{1^T c_{-j}} \in [0, Cc^L]$, then c_j^* is the solution to

   $$(c_j^* + \mathbf{1^T c_{-j}})^{-\alpha} - \alpha c_j^* (c_j^* + \mathbf{1^T c_{-j}})^{-\alpha-1} - \tfrac{\hat{a}}{C - c_j^*} = 0; \tag{6.30}$$

   If $\mathbf{1^T c_{-j}} \in [Cc^L, C]$, then $c_j^* = C - \mathbf{1^T c_{-j}}$.

2. *The case with $\hat{a} > C^{1-\alpha}$:*

   If $\mathbf{1^T c_{-j}} \in [0, c^H]$, then c_j^* is the solution to

   $$(c_j^* + \mathbf{1^T c_{-j}})^{-\alpha} - \alpha c_j^* (c_j^* + \mathbf{1^T c_{-j}})^{-\alpha-1} - \tfrac{\hat{a}}{C - c_j^*} = 0; \tag{6.31}$$

   If $\mathbf{1^T c_{-j}} \in [c^H, C]$, then $c_j^* = 0$.

The proof of Proposition 6.3.3 is given in Appendix 6.5.5. As expected, the Nash equilibrium of the spectrum opportunity allocation game depends on $\hat{a}$, α, and the number of PUs. We have the following necessary and sufficient condition for the uniqueness of Nash equilibrium.

Proposition 6.3.4. *The Nash equilibrium of the spectrum opportunity allocation game is unique if and only if* $c^L > \frac{|\mathcal{N}|-1}{|\mathcal{N}|}$, *and at the spectrum opportunity equilibrium,* $c_j^* = c^*$, $\forall j \in \mathcal{N}$, *where* c^* *is the solution to*

$$(|\mathcal{N}|c^*)^{-\alpha}(1 - \tfrac{\alpha}{|\mathcal{N}|}) = \tfrac{\hat{a}}{C-c^*}. \tag{6.32}$$

The proof is given in Appendix 6.5.6. Note that there exists a *threshold* for the number of PUs, denoted by N_{PU}, in the case with $0 < c^L < 1$, where N_{PU} is given by $\lfloor \frac{1}{1-c^L} \rfloor$[9]. Accordingly, Proposition 6.3.4 can be treated as a criterion for the PUs to decide whether to join in the competition or not, because each PU can calculate the pricing equilibrium when it gathers the necessary information based on Proposition 6.3.4. In the case with $0 < c^L < 1$, it needs to check whether the condition $N_{PU} > |\mathcal{N}|$ holds. This is because if $N_{PU} \leq |\mathcal{N}|$, the pricing equilibrium will be $p_{\min}$, which indicates that it is unprofitable to sell spectrum to SUs due to the strong competition.

6.3.3 An Algorithm for Computing Nash Equilibria

To achieve the Nash equilibrium of the dynamic game, we present an iterative algorithm for each PU. Based on Proposition 6.3.1, if $\sum_{j\in\mathcal{N}} c_j > C$, the spectrum allocation is inefficient, i.e., there always exists some PU whose supply is larger than the demand. Thus, each PU first updates its spectrum opportunity based on the demand to fully utilize its spectrum. After the necessary condition $\sum_{j\in\mathcal{N}} c_j \leq C$ is satisfied, each PU can update its spectrum opportunity in the Stage I based on Proposition 6.3.3. We assume that the "total budget" G of SUs is available to each PU. The proposed algorithm for computing the market equilibrium is summarized in Algorithm 6. Based on Proposition

[9] $\lfloor x \rfloor$ denotes the largest integer not greater than x.

6.3.1, Proposition 6.3.3 and Proposition 6.3.4, Algorithm 6 provably converges to the equilibrium of the game, as also verified by simulation.

Algorithm 6 Computing the Nash equilibrium of the multiple PU market

Initialization: Each PU collects the budget information of SUs, i.e., $\{\sigma_i\}$.
At the beginning of each period
1) **If** $\sum_{j\in\mathcal{N}} c_j > C$ **then**
Each PU sets $c_j = eD_j(p_j, \mathbf{p_{-j}})$, and broadcasts c_j.
Else
Each PU sets c_j based on Proposition 6.3.3, and broadcasts c_j.
Endif
2) Each PU sets $p_j = \max\left(p_{\min}, \left(\frac{eG}{\sum_{j\in\mathcal{N}} c_j}\right)^{\alpha}\right)$, and broadcasts p_j.
3) Each SU randomly chooses a PU's channel from the set of PUs with the lowest price in $\mathcal{N}$ with equal probability.
4) Each PU admits new SUs when $\frac{c_j}{e} > D_j(p_j, \mathbf{p_{-j}})$.

Remarks: Algorithm 6 is applicable to the scenarios where the PUs can vary their spectrum opportunities. When the spectrum opportunities are fixed, the three-stage game reduces to a two-stage game without the stage of spectrum opportunity allocation. In this case, the equilibrium of the game is given by Proposition 6.3.1.

6.3.4 Numerical Examples: Equilibria for Competitive PUs

In this section, we examine the Nash equilibrium of the three-stage game in the market with homogeneous PUs. First, we illustrate the existence and the uniqueness of the Nash equilibrium for two PUs, in the case with $c^L > 0.5$. Then, we consider a more general system model with four PUs and examine the convergence performance of Algorithm 6. In the end, we demonstrate how the equilibrium price evolves under different elasticities of SUs and different numbers of PUs. In each experiment, C is equal to 20, and each SU's budget σ_i is generated uniformly in the interval $[0, 4]$, and is fixed for different α for the sake of comparison.

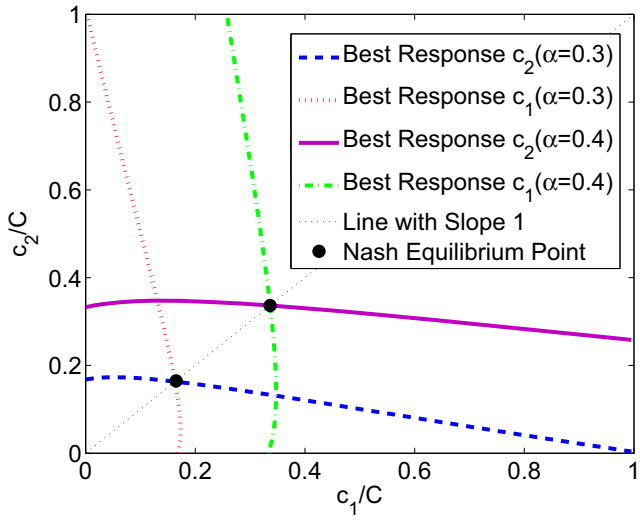

Figure 6.7: Illustrating the existence and uniqueness of spectrum opportunity equilibrium.

The existence and the uniqueness of the Nash equilibrium, corresponding to the competitive spectrum opportunity of two PUs, is illustrated in Fig. 6.7. Based on Proposition 6.3.4, when $c^L > 0.5$, there exists a unique Nash equilibrium, as verified in Fig. 6.7. In particular, we change the inverse elasticity (i.e., α) of SUs from 0.3 to 0.4 in order to show how the spectrum opportunity equilibrium evolves. We observe that the spectrum equilibrium lies on the line with slope one, due to the symmetry of the best response functions, and increases with α, since the SUs become more insensitive to prices, which motivates the PUs to offer more spectrum to the SUs.

Next, we examine the convergence performance of Algorithm 6 in the case of four PUs. Here, we choose $\alpha = 0.3$ and $a = 30$ such that $0 < \hat{a} \leq C^{1-\alpha}$ and $c^L > 0.75$. As illustrated in Fig. 6.8, the sum of initial normalized spectrum opportunities[10] is greater than 1, i.e., $\sum_{j\in\mathcal{N}} c_j > C$, in which case there is no equilibrium point based on Proposition 6.3.1. Each PU then updates

[10]The spectrum opportunity of each PU c_j is normalized by C.

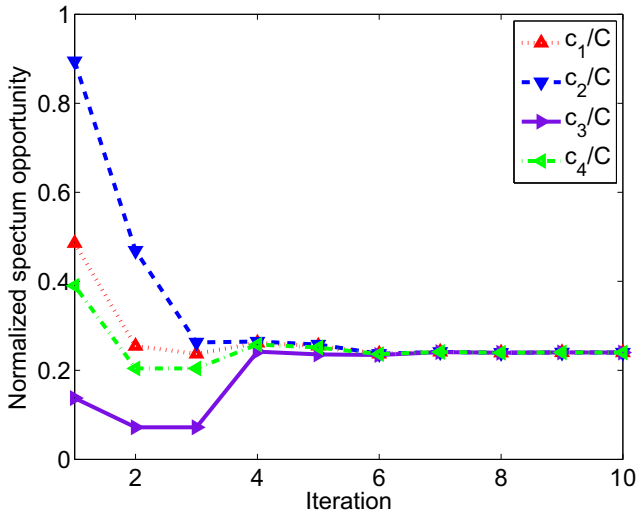

Figure 6.8: Convergence of spectrum opportunity equilibrium.

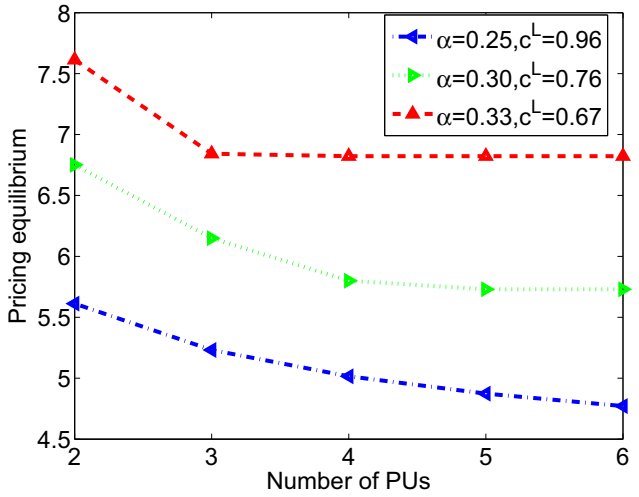

Figure 6.9: Price equilibrium corresponding to different elasticities of SUs and different numbers of PUs.

its offered spectrum opportunity based on its current demand (this process corresponds to the iterations from 1 to 3 in Fig. 6.8). Once the total supply is within the feasible region, each PU adjusts its supply based on Proposition 6.3.3. According to Proposition 6.3.4, there exists a unique Nash equilibrium,

which is further verified in Fig. 6.8.

Fig. 6.9 depicts how the equilibrium price evolves under different elasticities of SUs and different numbers of PUs. Specifically, we choose $a = 30$ and $|\mathcal{M}| = 200$. As expected, the equilibrium price increases with α. For each α, the equilibrium price decreases with the number of PUs, due to more competition among the PUs. In other words, the SUs can benefit from the competition among the PUs. Moreover, the equilibrium price approaches $p_{\min}$ as the number of PUs increases.

6.4 Conclusion

This chapter studied pricing-based decentralized spectrum access in cognitive radio networks, where SUs compete via random access for available spectrum opportunities. We developed two models: one with the monopoly PU market and the other with the multiple PU market. For the monopoly PU market model, we applied the revenue maximization approach to characterize the appropriate choice of flat and usage prices, and derived a Pareto optimal solution, which was shown to be near-optimal. More importantly, this Pareto optimal solution exhibits a decentralized structure, i.e., the Pareto optimal pricing strategy and access probabilities can be computed by the PU and the SUs locally. We also analyzed a PU profit maximization problem by examining the tradeoff between the PU's utility and its revenue.

We then studied the multiple PU market model by casting the competition among PUs as a three-stage Stackelberg game, in terms of access prices and the offered spectrum opportunities. We showed the existence of the Nash equilibrium, and derived a necessary and sufficient condition for the unique-

ness of Nash equilibrium for the case with homogeneous PUs. Intuitively, this condition can be used by PUs to decide whether to join in the competition or not, i.e., when the number of PUs grows larger than a certain threshold, the competition among PUs is too strong, indicating that it is unprofitable for a PU to sell spectrum to SUs. Then we developed an iterative algorithm for strategy adaption to achieve the Nash equilibrium.

It remains open to characterize the condition for the uniqueness of Nash equilibrium for the case with heterogeneous PUs. Another interesting direction is to investigate transient behaviors corresponding to dynamic spectrum access, in the presence of spectrum hole dynamics.

6.5 Appendix

6.5.1 Proof of Lemma 6.2.2

The proof of the first property is contained in that for the second one, which can be derived from the Lagrangian of (6.8) by utilizing the Karush-Kuhn-Tucker (KKT) conditions. Specifically, the Lagrangian of (6.8) is given by

$$\begin{aligned} L(\mathbf{z},\mathbf{p},\lambda) &= \textstyle\sum_{i\in\mathcal{M}}(p_i-\lambda_i)d_i(p_i) \\ &\quad +c\textstyle\sum_{i\in\mathcal{M}}\lambda_i z_i\prod_{j\neq i}(1-z_j). \end{aligned}$$

By the KKT conditions, for optimal $\mathbf{z}^*$ and $\mathbf{p}^*$, the system must satisfy the following equations:

$$\begin{aligned} &\tfrac{\partial L}{\partial z_i} = c\lambda_i\textstyle\prod_{j\neq i}(1-z_j) \\ &-c\textstyle\sum_{k\neq i}\lambda_k z_k\prod_{j\neq i,j\neq k}(1-z_j)=0, \forall i\in\mathcal{M}. \end{aligned}$$

Based on Theorem 1 in [138], the solution to the above equations is

$$z_i = \tfrac{\sum_{k\neq i}\lambda_k-(|\mathcal{M}|-2)\lambda_i}{\sum_{k\in\mathcal{M}}\lambda_k}, \forall i\in\mathcal{M}. \tag{6.33}$$

It follows that

$$\sum_{i\in\mathcal{M}} z_i = 1.$$

To prove the converse for the second property, note that the set of z_i given in (6.33) is a stationary point for the function $L(\cdot)$. It is straightforward to see that the set of z_i given in (6.33) cannot achieve a minimum point of the function $L(\cdot)$. Hence, the set of z_i given in (6.33) must maximize $L(\cdot)$.

6.5.2 Proof of Lemma 6.2.3

The Lagrangian of (6.9) is given by

$$L(\mathbf{p},\lambda) = \sum_{i\in\mathcal{M}} p_i d_i(p_i) - \lambda(\sum_{i\in\mathcal{M}} d_i(p_i) - c\kappa).$$

By the KKT conditions, the optimal $\mathbf{p}^*$ of the system must satisfy the following equations:

$$\frac{\partial L}{\partial p_i^*} = d_i(p_i^*) + p_i^* d_i'(p_i^*) - \lambda d_i'(p_i^*) = 0, \forall i \in \mathcal{M}.$$

The above equations can be written as

$$1 - \frac{\lambda}{p_i^*} = -\frac{d_i(p_i^*)}{p_i^* d_i'(p_i^*)} = \alpha, \forall i \in \mathcal{M},$$

where $\frac{1}{\alpha} = -\frac{p_i \hat{s}_i'(p_i)}{\hat{s}_i(p_i)}$ is the elasticity of SUs. Therefore, the optimal prices $p_i^* = \frac{\lambda}{1-\alpha}, \forall i \in \mathcal{M}$ are the same. Further based on the constraint of (6.9), $\sum_{i\in\mathcal{M}} d_i(p_i) = c\kappa$, the optimal price can be derived as (6.11) by substituting the demand function $d_i(p_i)$ for (6.4).

6.5.3 Proof of Lemma 6.2.4

Since (6.13) is strictly convex, we can solve it by first considering its dual problem. The Lagrangian of (6.13) is given by

$$\begin{aligned} L(\kappa', \mathbf{z}) &= \kappa' + \sum_{i\in\mathcal{M}} \lambda_i (\log z_i \\ &\quad + \sum_{j\neq i} \log(1 - z_j) - \log w_i - \kappa'). \end{aligned}$$

By the KKT conditions, we have $\sum_{i\in\mathcal{M}}\lambda_i = 1$ and $z_i = \lambda_i$. Thus, the solution of the dual problem satisfies the Pareto optimal condition, i.e., $\sum_{i\in\mathcal{M}} z_i = 1$, given by Lemma 6.2.2.

The dual problem can be written as:

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{M}}\lambda_i(\log\lambda_i + \sum_{j\neq i}\log(1-\lambda_j) - \log w_i) \\ \text{subject to} & \sum_{i\in\mathcal{M}}\lambda_i = 1, \\ & 0 \leq \lambda_i \leq 1, i \in \mathcal{M}, \\ \text{variables} & \{\lambda\}. \end{array} \tag{6.34}$$

By manipulating the summations and utilizing the constraint, the above problem can be written as

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{M}}(\lambda_i \log\frac{\lambda_i}{w_i} + (1-\lambda_i)\log(1-\lambda_i)) \\ \text{subject to} & \sum_{i\in\mathcal{M}}\lambda_i = 1, \\ & 0 \leq \lambda_i \leq 1, i \in \mathcal{M}, \\ \text{variables} & \{\lambda\}. \end{array} \tag{6.35}$$

Since (6.35) is strictly convex, therefore the optimal solution can be derived by the KKT conditions, which imply that $\lambda_i = \frac{w_i}{w_i + e^{-u}}$, where u is given by (6.15).

When the number of SUs is large, let $w_{\min} = \sigma_{\min}^{\frac{1}{\alpha}}/G$, where $\sigma_{\min} = \min_{i\in\mathcal{M}}\sigma_i$. By (6.15), we have

$$\begin{aligned} 1 &= \sum_{i\in\mathcal{M}}\frac{1}{1+w_i^{-1}e^{-u}} \\ &\leq |\mathcal{M}|\frac{1}{1+w_{\min}^{-1}e^{-u}}. \end{aligned}$$

Then we have $e^{-u} \geq w_{\min}(|\mathcal{M}|-1)$. For any z_i, we have

$$\begin{aligned} z_i = \frac{1}{1+w_i^{-1}e^{-u}} &\leq \frac{1}{1+w_i^{-1}w_{\min}(|\mathcal{M}|-1)} \\ &= \frac{1}{1+(\frac{\sigma_{\min}}{\sigma_i})^{\frac{1}{\alpha}}(|\mathcal{M}|-1)} \\ &\to 0, \text{as } |\mathcal{M}| \to \infty. \end{aligned}$$

Thus $\max_{i\in\mathcal{M}} z_i \to 0, \text{as } |\mathcal{M}| \to \infty$. Since $\sum_{i\in\mathcal{M}} z_i = 1$, we have $\prod_{i\neq j}(1 - z_j) \to \frac{1}{e}$. Therefore, Lemma 6.2.4 is proved.

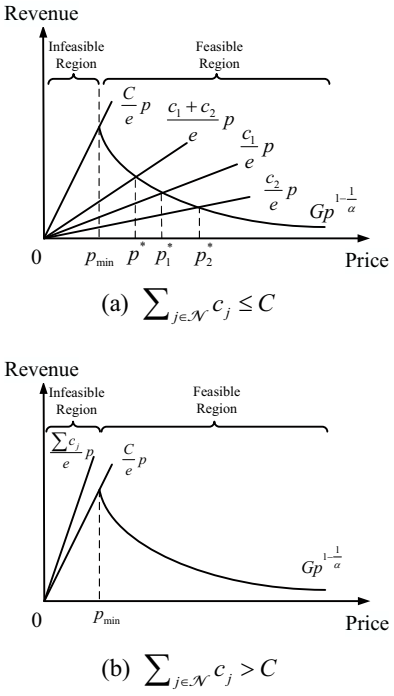

Figure 6.10: PU's Revenue at different prices and spectrum opportunities.

6.5.4 Proof of Proposition 6.3.1

We first show that $\sum_{j\in\mathcal{N}'} c_j \le C$ is the sufficient condition for the existence of the Nash equilibrium price, and that under that condition, the equilibrium price is uniquely determined by (6.27).

First, we consider the case where there are only two PUs. We show that there does not exist a pricing equilibrium such that $p_1^* \neq p_2^*$. Suppose that PU 1 and PU 2 have positive spectrum opportunities c_1 and c_2. Without loss of generality, we assume that $c_1 \ge c_2$. The optimal prices for PU 1 and PU 2 are shown in Fig. 6.10, where the curve $Gp^{1-\frac{1}{\alpha}}$ represents the optimal revenue that each PU can earn when there is no competition. Since it is only

when the demand equals to the supply that each PU can achieve its Pareto optimal revenue, the optimal price for each PU is at the intersection point of the curve and the line shown in Fig. 6.10. Obviously, $p_1^* \leq p_2^*$ when $c_1 \geq c_2$. From the analysis of Stage I, we know that SUs will choose PU 1 which makes PU 2 have no revenue. In this case, PU 2 will decrease its price at least less than or equal to PU 1 to get some revenue. And the price reduction will not end until both PUs announce the same price.

Next, we show that the equilibrium price is at the point p^* as shown in Fig. 6.10. Suppose that the equilibrium price p'^* is not at the point p^*. In this case, the total demand is not equal to the total supply. Thus, at least one PU's demand is not equal to its supply. For the case $p'^* > p^*$, the total demand is less than the total supply, which means at least one PU's demand is less than its supply. Without loss of generality, we assume that the PU 1's demand is less than its supply. Based on Lemma 6.2.2, PU 1 will decrease its price to make its demand equal to its supply, which will also make PU 2 decrease its price to achieve pricing equilibrium. Thus, when $p'^* > p^*$, both PUs will decrease their prices. For the other case $p'^* < p^*$, the total demand is larger than the total supply, which means at least one PU's demand is larger than its supply. Without loss of generality, we assume that the PU 1's demand is larger than its supply. Based on (6.26), PU 1 will increase its price to make its demand equal to its supply. In this case, the price of PU 1 will be larger than the price of PU 2, which makes all SUs choose PU 2. Therefore, the demand of PU 2 will be larger than its supply, which forces PU 2 to increase its price to achieve more revenue. Thus, when $p'^* < p^*$, both PUs will increase their prices. Hence, the equilibrium price is at p^* which can be determined by (6.27). Since $p^* \in [p_{\min}, \infty)$, the total supply $\sum_{j \in \mathcal{N}} c_j$ is less than or equal to

C.

Thus far, we have discussed the case for two PUs. The above results can be easily generalized to the case with more than two PUs by following similar steps. Therefore, the equilibrium price is uniquely determined by (6.27), when $\sum_{j\in\mathcal{N}} c_j \leq C$.

To show that $\sum_{j\in\mathcal{N}'} c_j \leq C$ is the necessary condition for the existence of the Nash equilibrium price, it suffices to show that no equilibrium exists when $\sum_{j\in\mathcal{N}} c_j > C$. By the definition of **Game at Stage II**, each PU can choose a price p_j from the feasible set $\mathcal{P} = [p_{\min}, \infty)$, where $p_{\min}$ is determined by (6.10) at $c = C$. When $\sum_{j\in\mathcal{N}} c_j > C$, the equilibrium price would be smaller than $p_{\min}$, which means that the total supply is greater than the total demand. In other words, some PU's spectrum opportunities are unused. Thus, those PUs can always decrease the supplied spectrum to improve their own utilities and make the demand equal to their supply in the end. Intuitively speaking, since the revenue curve has no intersection point with the line with slope $\sum_{j\in\mathcal{N}'} c_j > C$ as shown in Fig. 6.10, this means that in this region there is no equilibrium point. Therefore, the necessary condition for PUs to achieve the Nash equilibrium is $\sum_{j\in\mathcal{N}} c_j \leq C$.

6.5.5 Proof of Proposition 6.3.3

Due to the concavity of $\hat{R}_j(c_j, \mathbf{c_{-j}})$, we can obtain the best response function by checking the first order condition and boundary conditions. The first order condition is

$$\begin{aligned}\frac{\partial \hat{R}_j(c_j,\mathbf{c_{-j}})}{\partial c_j} &= (c_j + \mathbf{1^T c_{-j}})^{-\alpha} - \alpha c_j (c_j + \mathbf{1^T c_{-j}})^{-\alpha-1} \\ &\quad - \frac{\hat{a}}{C - c_j},\end{aligned}$$

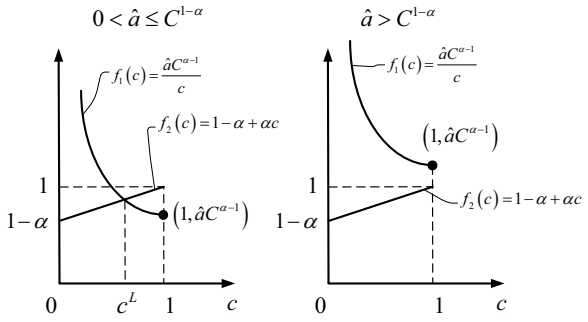

Figure 6.11: Decision threshold c^L.

and the boundary conditions can be written as

$$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j=0} = (\mathbf{1^T c_{-j}})^{-\alpha} - \frac{\hat{a}}{C}, \tag{6.36}$$

$$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j=C-\mathbf{1^T c_{-j}}} = C^{-\alpha}(1-\alpha+\frac{\alpha(\mathbf{1^T c_{-j}})}{C} - \frac{\hat{a}C^{\alpha}}{\mathbf{1^T c_{-j}}}), \tag{6.37}$$

both of which depend on its competitors spectrum opportunities and the parameter $\hat{a}$. For different $\mathbf{c_{-j}}$ and $\hat{a}$, the jth PU's best response strategy can be written as follows:

1. Case $0 < \hat{a} \le C^{1-\alpha}$:

   $$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j=0} \ge 0,$$

   i.e., the jth PU would sell spectrum opportunity to the SUs. Then the optimal spectrum opportunity c_j^* depends on the boundary condition (6.37). From Fig. 6.11, we know that there exists a decision threshold $c^L = \frac{1}{2\alpha}(\alpha - 1 + \sqrt{(\alpha-1)^2 + 4\alpha\hat{a}C^{\alpha-1}})$ for PU j, which is the solution to

   $$1 - \alpha + \alpha c^L = \frac{\hat{a}C^{\alpha-1}}{c^L}.$$

   Based on its competitors' spectrum opportunities, the jth PU's decision is outlined as follows:

a) $\mathbf{1^T c_{-j}} \in [0, Cc^L]$:

$$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j = C - \mathbf{1^T c_{-j}}} \leq 0.$$

The best response strategy of the jth PU is determined by its first order condition (6.30).

b) $\mathbf{1^T c_{-j}} \in [Cc^L, C]$:

$$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j = C - \mathbf{1^T c_{-j}}} \geq 0.$$

The best response strategy of the jth PU is to sell as much spectrum opportunity as possible, i.e., $c_j^* = C - \mathbf{1^T c_{-j}}$.

2. Case $\hat{a} > C^{1-\alpha}$:

From Fig. 6.11, we know

$$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j = C - \mathbf{1^T c_{-j}}} \leq 0.$$

The optimal spectrum opportunity c_j depends on the boundary condition (6.36), from which we know there exists a decision threshold $c^H = (\frac{\hat{a}}{C})^{-\frac{1}{\alpha}}$. Based on c^H and its competitors' spectrum opportunities, the decisions of the jth PU are as follows:

a) $\mathbf{1^T c_{-j}} \in [0, c^H]$:

$$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j = 0} \geq 0.$$

The best response strategy of the jth PU is determined by its first order condition (6.31).

b) $\mathbf{1^T c_{-j}} \in [c^H, C]$:

$$\frac{\partial \hat{R}_j(c_j, \mathbf{c_{-j}})}{\partial c_j}|_{c_j = 0} \leq 0.$$

The best response strategy of the jth PU is not to sell any spectrum opportunity, i.e., $c_j^* = 0$.

6.5.6 Proof of Proposition 6.3.4

Due to the concavity of $\hat{R}_j(c_j, \mathbf{c_{-j}})$ in c_j, the existence of equilibrium can be readily shown based on [137]. In what follows, we will derive the necessary and sufficient condition for the uniqueness of Nash equilibrium, based on the best response strategy.

1. Case $0 < \hat{a} \leq C^{1-\alpha}$:

   Define $\mathcal{N}_1 = \{j | c_j^* \text{ is the solution to (6.30)}, \forall j \in \mathcal{N}\}$ and $\mathcal{N}_2 = \{j | c_j^* = C - \mathbf{1^T c_{-j}}, \forall j \in \mathcal{N}\}$ as the sets of PUs choosing decision c_j^* as the solution to (6.30) and $c_j^* = C - \mathbf{1^T c_{-j}}$ respectively, where $|\mathcal{N}_1| + |\mathcal{N}_2| = |\mathcal{N}|$. Let $c_{j_1}^*, \forall j_1 \in \mathcal{N}_1$ and $c_{j_2}^*, \forall j_2 \in \mathcal{N}_2$ denote the spectrum opportunity equilibrium for PUs in the set $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively. Assume that both sets $\mathcal{N}_1$ and $\mathcal{N}_2$ are nonempty. Based on Proposition 6.3.3, we know $\sum_{j\in\mathcal{N}} c_j^* = C$, and $c_{j_1}^*$ is the solution of

   $$\begin{aligned} & C^{-\alpha} - \alpha c_{j_i}^* C^{-\alpha-1} - \tfrac{\hat{a}}{C - c_{j_i}^*} = 0 \\ \Leftrightarrow \quad & 1 - \alpha + \alpha \tfrac{C - c_{j_i}^*}{C} = \tfrac{\hat{a} C^{\alpha}}{C - c_{j_i}^*} \\ \Leftrightarrow \quad & 1 - \alpha + \alpha c^L = \tfrac{\hat{a} C^{\alpha-1}}{c^L}. \end{aligned}$$

   Thus $c_{j_1}^* = C(1 - c^L), \forall j_1 \in \mathcal{N}_1$. Since $C - c_{j_2}^* \geq C c^L, \forall j_2 \in \mathcal{N}_2$, we can get $\sum_{j_2\in\mathcal{N}_2} c_{j_2}^* \leq |\mathcal{N}_2| C(1 - c^L)$. Utilizing the condition $\sum_{j_2\in\mathcal{N}_2} c_{j_2}^* + |\mathcal{N}_1| C(1 - c^L) = C$ (i.e., $\sum_{j\in\mathcal{N}} c_j^* = C$), c^L needs to satisfy

   $$c^L \leq \tfrac{|\mathcal{N}| - 1}{|\mathcal{N}|}.$$

   Therefore, we can summarize the spectrum opportunity equilibria as follows:

   a) Case $c^L \leq \frac{|\mathcal{N}|-1}{|\mathcal{N}|}$:

      In this case, there exist infinitely many spectrum opportunity equi-

libria that satisfy

$$\sum_{j_2 \in \mathcal{N}_2} c_{j_2}^* = C - |\mathcal{N}_1| C(1 - c^L),$$

$$c_{j_1}^* = C(1 - c^L), \forall j_1 \in \mathcal{N}_1,$$

$$C - c_{j_2}^* \geq C c^L, \forall j_2 \in \mathcal{N}_2.$$

b) Case $c^L > \frac{|\mathcal{N}|-1}{|\mathcal{N}|}$:

In this case, we have either $\mathcal{N}_1 = \mathcal{N}$, $\mathcal{N}_2 = \emptyset$ or $\mathcal{N}_1 = \emptyset$, $\mathcal{N}_2 = \mathcal{N}$. For the case $\mathcal{N}_1 = \emptyset$, $\mathcal{N}_2 = \mathcal{N}$, we have $\sum_{j \in \mathcal{N}} c_j^* = C$ and $C - c_j^* \geq Cc^L, \forall j \in \mathcal{N}$, by which we have $\sum_{j \in \mathcal{N}} (C - c_j^*) \geq |\mathcal{N}| C c^L$, i.e., $|\mathcal{N}| - 1 \geq |\mathcal{N}| c^L$, which yields $1 \geq \frac{|\mathcal{N}|}{|\mathcal{N}|-1} c^L > 1$ due to $c^L > \frac{|\mathcal{N}|-1}{|\mathcal{N}|}$. Obviously, this contradicts the fact $1 = 1$. Hence, the only possible case is $\mathcal{N}_1 = \mathcal{N}$, $\mathcal{N}_2 = \emptyset$. Due to the homogeneity of best response function, there exists a unique spectrum opportunity equilibrium [139], i.e., $c_j^* = c^*, \forall j \in \mathcal{N}$, where c^* is the solution to

$$(|\mathcal{N}| c^*)^{-\alpha} - \alpha c^* (|\mathcal{N}| c^*)^{-\alpha - 1} = \tfrac{\hat{a}}{C - c^*}.$$

2. Case $\hat{a} > C^{1-\alpha}$:

Define $\mathcal{N}_1 = \{j | c_j^* \text{ is the solution to (6.31)}, \forall j \in \mathcal{N}\}$ and $\mathcal{N}_2 = \{j | c_j^* = 0, \forall j \in \mathcal{N}\}$ as the sets of PUs choosing decision c_j^* as the solution to (6.31) and $c_j^* = 0$ respectively, where $|\mathcal{N}_1| + |\mathcal{N}_2| = |\mathcal{N}|$. Let $c_{j_1}^*, \forall j_1 \in \mathcal{N}_1$ and $c_{j_2}^*, \forall j_2 \in \mathcal{N}_2$ denote the spectrum opportunity equilibrium for PUs in the set $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively. Based on Proposition 6.3.3, we know that $c_{j_2}^* = 0$, and we can use (6.31) to calculate $c_{j_1}^*$. Then $c_{j_1}^* = c^*, \forall j_1 \in \mathcal{N}_1$, where c^* is the solution to

$$(|\mathcal{N}_1| c^*)^{-\alpha} - \alpha c^* (|\mathcal{N}_1| c^*)^{-\alpha - 1} = \frac{\hat{a}}{C - c^*}.$$

After some algebra, we have $|\mathcal{N}_1|c^* < (\frac{\hat{a}}{C})^{-\frac{1}{\alpha}}$. Since for each PU in $\mathcal{N}_2$, $|\mathcal{N}_1|c^* < (\frac{\hat{a}}{C})^{-\frac{1}{\alpha}}$,i.e., $\mathbf{1^T c_{-j}} \in [0, c^H]$, the best response for PUs in $\mathcal{N}_2$ is the decision (6.31), which shows that all PUs will choose the decision (6.31). Due to the homogeneity of best response function, there exists a unique spectrum opportunity equilibrium [139], which can be determined by (6.32).

When $\hat{a} > C^{1-\alpha}$, we have $c^L > 1$ based on the proof of Proposition 6.3.3. Therefore, $c^L > \frac{|\mathcal{N}|-1}{|\mathcal{N}|}$. In summary, when $c^L > \frac{|\mathcal{N}|-1}{|\mathcal{N}|}$, there exists a unique Nash equilibrium. The other direction follows directly based on the above discussion of spectrum opportunity equilibria.

Chapter 7

DISTRIBUTED STOCHASTIC POWER CONTROL IN AD HOC NETWORKS: A NONCONVEX OPTIMIZATION CASE

7.1 Introduction

The broadcast nature of wireless transmissions makes wireless networks susceptible to interference, which deteriorates quality of service (QoS) provisioning. Power control is considered as a promising technique to mitigate interference. One primary objective of power control is to maximize the system utility that can achieve a variety of fairness objectives among users [7–10]. However, maximizing the system utility, under the physical interference model, often involves nonconvex optimization and it is known to be NP-hard, due to the complicated coupling among users through mutual interference effects [11].

Due to the nonconvex nature of the power control problem, it is challenging to find the globally optimal power allocation in a distributed manner. Notably, [12, 13, 140, 141] devised distributed power control algorithms to find power allocations that can only satisfy the local optimality conditions, but global optimality could not be guaranteed in general, except for some special convexifiable cases (e.g., with strictly increasing log-concave utility functions). Another thread of work applied game-theoretic approaches to power control by treating it as a non-cooperative game among transmitters [14, 15]. However, distributed solutions that converge to a Nash equilibrium may be suboptimal in terms of maximizing the total system utility. Different from these approaches, [142] transformed the power control problem into a DC (difference of convex functions) optimization problem [143]. Then, the global optimal solution

can be solved in a centralized manner with the branch-and-bound algorithm. Recent work [16] proposed a globally optimal power control scheme, named MAPEL, by exploiting the monotonic nature of the underlying optimization problem. However, the complexity and the centralized nature of MAPEL hinder its applicability in practical scenarios, and thus it can be treated rather as a benchmark for performance evaluation in distributed networks.

To find the globally optimal power allocation in a distributed setting, recent work [17] has proposed the SEER algorithm based on Gibbs sampling [18], which can approach the globally optimal solution in an asymptotic sense when the control parameter in Gibbs sampling tends to infinity. Notably, for each iteration in the SEER algorithm, each user utilizes Gibbs sampling to compute its transition probability distribution for updating its transmission power, where the requirement for message passing and computing the transition probability distribution in each iteration can be demanding when applied to ad hoc communications without centralized control.

A challenging task in distributed power control in ad hoc networks is to reduce the amount of message passing while preserving the global optimality. To tackle this challenge, we first show that the globally optimal point lies on the boundary of the feasible region. This property is utilized to transform the utility maximization problem into an equivalent max-min problem with more structure, which can be solved by combining recent advances in extended duality theory (EDT) [19] with simulated annealing (SA) [20]. Compared with the classical duality theory with nonzero duality gap for nonconvex optimization problems, EDT can guarantee zero duality gap between the primal and dual problems by utilizing nonlinear Lagrangian functions. This property allows for solving the nonconvex problem by its *extended dual* while preserving the

global optimality with distributed implementation. Furthermore, as will be shown in Section 7.2, for the subproblem of each individual user, the extended dual can then be solved through stochastic search with SA. In particular, we first transform the original utility maximization problem into an equivalent max-min problem. This step is based on the key observation that in the case with continuous and strictly increasing utility functions, the globally optimal solution is always on the *boundary* of the feasible (utility) region. Then, appealing to EDT and SA, we develop a distributed stochastic power control (DSPC) algorithm that stochastically searches for the optimal power allocation in the neighborhood of the feasible region's boundary, instead of bouncing around in the entire feasible region.

Specifically, we first show that DSPC can achieve the global optimality in the underlying nonconvex optimization problem, although the convergence rate can be slow (but this is clearly due to the slow convergence nature of SA with logarithmic cooling schedule). Then, to improve the convergence rate of DSPC, we propose an enhanced DSPC (EDSPC) algorithm that employs the geometric cooling schedule [144] and performs a careful selection of penalty parameters. As a benchmark for performance evaluation, we also develop a centralized algorithm to search for the globally optimal solution over simplices that cover the feasible region. The performance gain is further verified by comparing our distributed algorithms with MAPEL [16], SEER [17], and ADP [13] algorithms. Worth noting is that the proposed DSPC and EDSPC algorithms do not require any knowledge of channel gains, which is typically needed in existing algorithms, and instead they need only the standard feedback of Signal-to-Interference-plus-Noise (SINR) for adaptation.

Next, we integrate the proposed distributed power control approach

with the back-pressure algorithm [145] and devise a joint scheduling and power allocation policy for improving the queue stability in the presence of dynamic packet arrivals and departures. This policy fits into the dynamic back-pressure and resource allocation framework and enables distributed utility maximization under stochastic packet traffic [146] [147]. Then, we generalize the study to consider multicast communications, where a single transmission may simultaneously deliver packets to multiple recipients [148, 149]. Specifically, we extend DSPC and EDSPC algorithms to multicast communications with distributed implementation, and show that these algorithms can also achieve the global optimality in terms of jointly maximizing the minimum rates on bottleneck links in different multicast groups.

The rest of the chapter is organized as follows. In Section 7.2, we first introduce the system model, establish the equivalence between the utility maximization problem and its max-min form, and then develop both centralized and distributed algorithms for the max-min problem. Next, building on these power control algorithms, we develop in Section 7.3 a joint scheduling and power allocation policy to stabilize queueing systems. The generalization to multicast communications is presented in Section 7.4. We conclude the chapter in Section 7.5.

7.2 Power Control for Unicast Communications

7.2.1 System Model

We consider an ad hoc wireless network with a set $\mathcal{L} = \{1, ..., L\}$ of links, where the channel is interference-limited, and all L links treat interference as noise, as illustrated in Fig. 7.1. Such a model of communication is readily applicable to cellular networks [7]. Each link consists of a dedicated transmitter-receiver

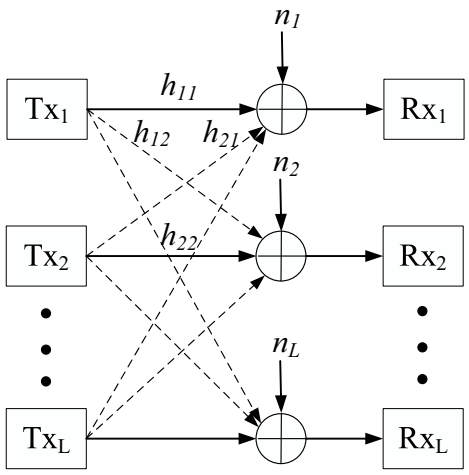

Figure 7.1: System model.

pair.[1] We denote by h_{lk} the fixed channel gain between user l's transmitter and user k's receiver, and by p_l the transmission power of link l with $P_l^{\max}$ being its maximum power constraint. It follows that the received SINR for the lth user with a matched filter receiver is given by

$$\gamma_l(\mathbf{p}) = \frac{h_{ll}p_l}{n_l + \sum_{k \neq l} h_{kl}p_k}, \tag{7.1}$$

where $\mathbf{p} = (p_1, ..., p_L)$ is a vector of the users' transmission powers and n_l is the noise power. Accordingly, the lth user receives the utility $U_l(\gamma_l)$, where $U_l(\cdot)$ is continuous and strictly increasing. We assume that each user l's utility is zero when $\gamma_l = 0$, i.e., $U_l(0) = 0$. For ease of reference, the key notation in this chapter is listed in Table 7.1.[2]

[1] We use the terms "user" and "link" interchangeably throughout the chapter.

[2] We use bold symbols (e.g., $\mathbf{p}$) to denote vectors and calligraphic symbols (e.g., $\mathcal{L}$) to denote sets.

Table 7.1: Summary of the key notation and definitions.

Notation	Definition
$\mathcal{L}$	set of links
L	total number of links
h_{lk}	channel gain from link l's transmitter to link k's receiver
$\mathbf{H}$	link gain matrix
p_l (in vector $\mathbf{p}$)	transmission power of link l
n_l (in vector $\mathbf{n}$)	noise power for link l
γ_l	SINR of link l
$\gamma_l(\cdot)$	SINR function of link l
$U_l(\cdot)$	utility function of link l
x_l (in vector $\mathbf{x}$)	ratio of link l's utility to the total network utility
r_l (in vector $\mathbf{r}$)	transmission rate of link l
$r_l(\cdot)$	transmission rate function of link l
α, β	penalty multipliers

7.2.2 Network Utility Maximization

We seek to find the optimal power allocation $\mathbf{p}^*$ that maximizes the overall system utility subject to the individual power constraints, given by the following optimization problem[3]:

$$\begin{array}{ll} \text{maximize} & \sum_{l\in\mathcal{L}} U_l(\gamma_l(\mathbf{p})) \\ \text{subject to} & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} & \{\mathbf{p}\}. \end{array} \tag{7.2}$$

In general, (7.2) is a nonconvex optimization problem[4]. In particular, if the utility function is the Shannon rate achievable over Gaussian flat fading channels, namely $U_l(\gamma_l(\mathbf{p})) = w_l \log(1+\gamma_l(\mathbf{p}))$, where $w_l > 0$ is a weight associated with user l, (7.2) boils down to the weighted sum rate maximization

[3]The QoS constraint for each link can be incorporated in (7.2), and the proposed algorithms in the following section can be easily adapted to this case at the cost of added notational complexity.

[4]For some special utility functions $U_l(\cdot)$, (7.2) can be transformed into a convex problem [9]. In this chapter, we focus on the nonconvex case that cannot be transformed to a convex problem by change of variables.

Table 7.2: The performance of the existing approaches for Case I and II.

Approach	Case I		Case II	
	Power	Sum Rate	Power	Sum Rate
GP	[20, 7.68]	3.02	[1, 0.61]	0.98
ADP	[20, 6.46]	3.10	[1, 2]	1.16
MAPEL	[20, 6.79]	3.10	[0, 2]	1.22
SEER	[20, 6.90]	3.10	[0, 2]	1.22

problem, which is known to be nonconvex and NP-hard [11]. Note that the weights in (7.2) can serve as the fairness measures [150] for different scenarios. In particular, in queueing systems, packet queues for arrival rates within the stability region can be stabilized by solving this weighted sum rate maximization problem, where the instantaneous queue lengths are chosen as the weights. In Section 7.3, we will discuss how to stabilize the packet queues by integrating our distributed power control algorithms with the back-pressure algorithm.

Let $\mathcal{F}$ denote the feasible utility region, where for each point $\mathbf{U} = (U_1, ..., U_L)$ in $\mathcal{F}$, there exists a power vector $\mathbf{p}$ such that $U_l = U_l(\gamma_l(\mathbf{p}))$ for all $l \in \mathcal{L}$. The feasible utility region $\mathcal{F}$ is nonconvex, and in general, finding the globally optimal solution to (7.2) in $\mathcal{F}$ is challenging. In the following example, we illustrate the geometry of $\mathcal{F}$ for the utility function $U_l(\gamma_l(\mathbf{p})) = w_l \log(1 + \gamma_l(\mathbf{p}))$ and evaluate the solutions to (7.2) given by some existing power control approaches discussed in Section 7.1.

Example: For the case with two links, Fig. 7.2 illustrates the nonconvex feasible utility region $\mathcal{F}$ for different system parameters. We compare the performance of the existing approaches [7, 13, 16, 17] in Table 7.2.

Remarks: The solutions to (7.2) given by [7, 13, 16] are either distributed but suboptimal or optimal but centralized. In particular, [7] solves

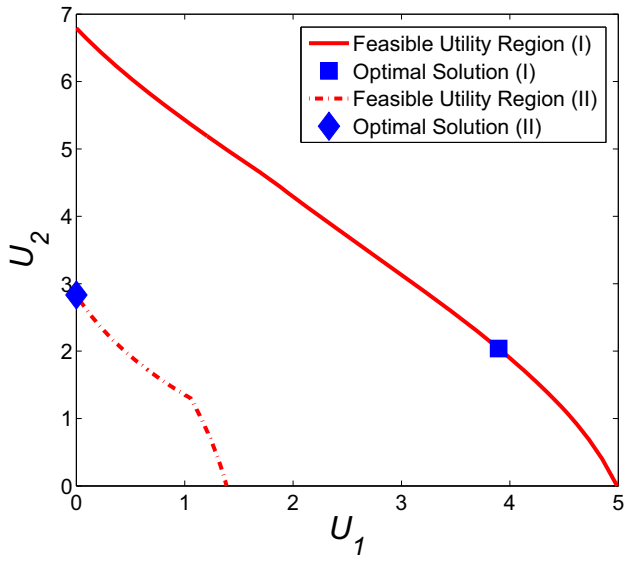

Figure 7.2: The feasible utility region $\mathcal{F}$. Case (I): the channel gains are given by $h_{11} = 0.73$, $h_{12} = 0.04$, $h_{21} = 0.03$, and $h_{22} = 0.89$, and the maximum power are $P_1^{\max} = 20$, $P_2^{\max} = 100$; Case (II): the channel gains are given by $h_{11} = 0.30$, $h_{12} = 0.50$, $h_{21} = 0.03$, and $h_{22} = 0.80$, and the maximum power are $P_1^{\max} = 1$, $P_2^{\max} = 2$. In both cases, the noise power is 0.1 for each link, and the weights are $w_1 = 0.57$, $w_2 = 0.43$.

(7.2) by using geometric programming (GP) under the high-SINR assumption, which yields a suboptimal solution to (7.2) when this assumption does not hold (e.g., this is the case in the example above). The ADP algorithm [13] can guarantee only local optimality[5] in a distributed manner. The MAPEL algorithm [16] can achieve the globally optimal solutions but it is centralized with high computational complexity. Compared with these algorithms, the SEER algorithm [17] can guarantee global optimality in a distributed manner but message passing needed in each iteration can be demanding, i.e., each link needs the knowledge of the channel gains, the receiver SINR and the signal power of all the other links. It is worth noting that the performance of SEER hinges heavily on the control parameter that can be challenging to choose on

[5]The local optimal solution found by ADP matches the globally optimal solution only in one of the cases that are illustrated in Table 7.2.

the fly.

7.2.3 From Network Utility Maximization to Minimum Weighted Utility Maximization

In order to devise low-complexity distributed algorithms that can guarantee global optimality, we first study the basic properties for the solutions to (7.2), before transforming (7.2) into a more structured max-min problem.

Lemma 7.2.1. *The optimal solution to (7.2) is on the boundary of the feasible utility region $\mathcal{F}$.*

Proof. Let $\mathbf{U}^*$ denote a globally optimal solution to (7.2) over $\mathcal{F}$, and γ^* denote the corresponding SINR that supports $\mathbf{U}^*$. Since $U_l(\cdot)$ is continuous and strictly increasing, proving that $\mathbf{U}^*$ is on the boundary of $\mathcal{F}$ is equivalent to showing that γ^* is on the boundary of the feasible SINR region. Suppose that γ^* is not on the boundary of the feasible SINR region, which indicates that there exists some point $\hat{\gamma}$ such that $\hat{\gamma}_l \geq \gamma_l^*$ for all $l \in \mathcal{L}$ and $\hat{\gamma}_{l'} > \gamma_{l'}^*$ for some $l' \in \mathcal{L}$. Since $U_l(\cdot)$ for any $l \in \mathcal{L}$ is strictly increasing in γ_l, we have $U_l(\hat{\gamma}_l) \geq U_l(\gamma_l^*)$ for all $l \in \mathcal{L}$ and $U_l(\hat{\gamma}_{l'}) > U_l(\gamma_{l'}^*)$ for some $l' \in \mathcal{L}$, which contradicts the fact that γ^* is a globally optimal solution. Hence, Lemma 7.2.1 follows. □

Based on Lemma 7.2.1, if we can characterize the boundary of $\mathcal{F}$, then it is possible to solve (7.2) efficiently. Thus motivated, we first establish, by introducing a "contribution weight" for each user, the equivalence between (7.2) and the minimum weighted utility maximization problem.

Lemma 7.2.2. *Problem (7.2) is equivalent to the following minimum weighted*

utility maximization:

$$\begin{aligned} \text{maximize} \quad & \min_{l\in\mathcal{L}} \tfrac{U_l(\gamma_l(\mathbf{p}))}{x_l} \\ \text{subject to} \quad & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ & 0 \le x_l \le 1, \forall l \in \mathcal{L} \\ & \sum_{l\in\mathcal{L}} x_l = 1 \\ \text{variables} \quad & \{\mathbf{p}, \mathbf{x}\}. \end{aligned} \tag{7.3}$$

Proof. Let $t = \sum_{l\in\mathcal{L}} U_l(\gamma_l(\mathbf{p}))$ denote the total utility. Since $U_l(\cdot)$ is nonnegative, we define $x_l \in [0,1]$ as a ratio for the contribution of user l's utility to t. Therefore, $U_l(\gamma_l(\mathbf{p})) = tx_l$ and $\sum_{l\in\mathcal{L}} x_l = 1$. Then (7.2) can be rewritten as

$$\begin{aligned} \text{maximize} \quad & t \\ \text{subject to} \quad & t = \tfrac{U_l(\gamma_l(\mathbf{p}))}{x_l}, \forall l \in \mathcal{L} \\ & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ & 0 \le x_l \le 1, \forall l \in \mathcal{L} \\ & 0 \le t, \sum_{l\in\mathcal{L}} x_l = 1 \\ \text{variables} \quad & \{\mathbf{p}, \mathbf{x}, t\}. \end{aligned} \tag{7.4}$$

In order to maximize t, it suffices then to relax $t = \frac{U_l(\gamma_l(\mathbf{p}))}{x_l}$ in (7.4) as $t \le \frac{U_l(\gamma_l(\mathbf{p}))}{x_l},\ \forall l \in \mathcal{L}$, which is equivalent to $t \le \min_{l\in\mathcal{L}} \frac{U_l(\gamma_l(\mathbf{p}))}{x_l}$. Therefore, (7.4) can be treated as the hypograph form of (7.3), i.e., (7.4) and (7.3) are equivalent [42], thereby concluding the proof. □

By transforming (7.2) to this more structured max-min problem (7.3), the problem is reduced to finding a globally optimal $\mathbf{x}^*$, given which we can efficiently obtain a globally optimal solution, i.e., the tangent point of the hyperplane and $\mathcal{F}$, as illustrated in Fig. 7.3. Intuitively speaking, $\mathbf{x}$ represents

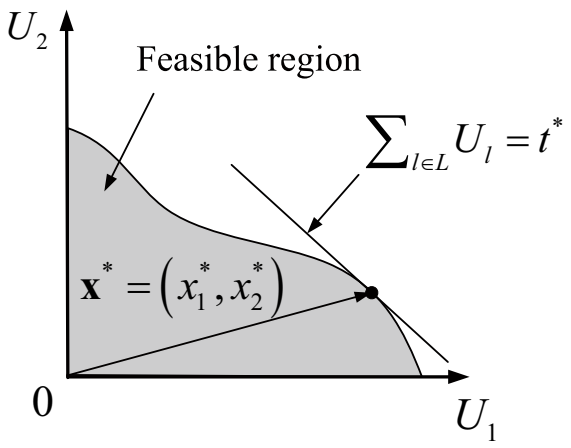

Figure 7.3: An illustration of the max-min problem for the case with two links.

a search direction. Once we find the best search direction $\mathbf{x}^*$, $\mathbf{p}^*$ can be obtained efficiently by searching along the direction of $\mathbf{x}^*$. Actually, for given $\mathbf{x}$, (7.3) is quasi-convex[6]. By introducing an auxiliary variable t, we obtain the following equivalent formulation:

$$\begin{aligned} \text{maximize} \quad & t \\ \text{subject to} \quad & U_l^{-1}(tx_l)(n_l + \sum_{k\neq l} h_{kl}p_k) \leq h_{ll}p_l \\ & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L},\ 0 \leq t \\ \text{variables} \quad & \{\mathbf{p}, t\}, \end{aligned} \tag{7.5}$$

which can be solved in polynomial time through binary search on t [42]. However, the optimal search direction $\mathbf{x}^*$ is difficult to find due to the nonconvex nature of the problem. In the following section, we study how to find the globally optimal search direction $\mathbf{x}^*$.

[6]By definition, a function $f : \mathbb{R}^n \to \mathbb{R}$ is quasi-convex, if its domain $\mathbf{dom}f$ and all its sublevel sets $\mathcal{S}_c = \{x \in \mathbf{dom}f | f(x) \leq c\}$, for $c \in \mathbb{R}$, are convex [42].

7.2.4 Centralized vs. Distributed Algorithms

In this section, we study algorithms achieving global optimality for (7.3). First, we propose a centralized algorithm for (7.3), which will serve as a benchmark for performance comparison. Then, by using EDT and SA, we propose a distributed algorithm, DSPC, for the problem (7.3). Building on this, we propose an enhanced algorithm EDSPC to improve the convergence rate of DSPC.

1) A centralized algorithm

Based on Lemma 7.2.1 and Lemma 7.2.2, we develop a centralized algorithm (Algorithm 7) to solve the max-min optimization problem (7.3) under consideration. Roughly speaking, by dividing the simplex $\mathcal{S} = \{\mathbf{x} | \sum_{l\in\mathcal{L}} x_l = 1, 0 \leq x_l \leq 1, \forall l \in \mathcal{L}\}$ into many small simplices, the algorithm can find the optimal point on the boundary of $\mathcal{F}$. Fig. 7.4 illustrates how the simplex cutting is performed for the case with three links. Compared with the MAPEL algorithm [16], Algorithm 7 directly computes the points on the boundary, instead of constructing a series of polyblocks to approximate the boundary of the feasible region.

Proposition 7.2.1. *Algorithm 7 converges monotonically to the globally optimal solution to (7.3) as the approximation factor ϵ approaches zero.*

Proof. For given ϵ, Algorithm 7 divides the simplex $\mathcal{S} = \{\mathbf{x} | \sum_{l\in\mathcal{L}} x_l = 1, 0 \leq x_l \leq 1, \forall l \in \mathcal{L}\}$ until the maximal diameter of the subdivided simplices δ^m is less than ϵ. Let $d(\epsilon)$ denote the minimum distance between the optimal solution $\mathbf{x}^*$ and the center point of the simplex that contains $\mathbf{x}^*$. Obviously, $d(\epsilon)$ is bounded by δ^m. Since δ^m decreases with the decreasing of ϵ, therefore

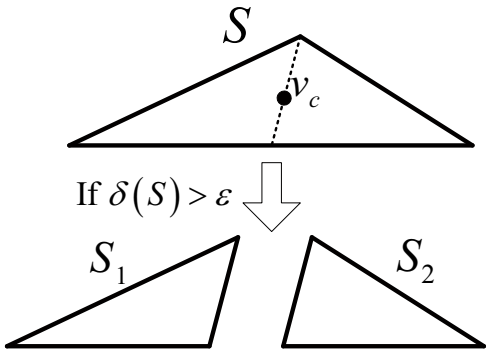

Figure 7.4: An illustration of the simplex cutting for the case with three links.

Algorithm 7

Initialization: Choose the approximation factor $\epsilon > 0$, and construct the initial simplex $\mathcal{S}$ with the vertex set $V = \{v_1, ..., v_L\}$, where $v_l = e_l$ and e_l is the lth unit coordinate vector. Let $v_c = \frac{1}{L}\sum_{l\in\mathcal{L}} v_l$ be the center of S. Compute $\mathbf{p}^*$ by solving (7.5) at the point $\mathbf{x} = v_c$. Denote $\delta(\mathcal{S}) = \max_{v\in V} |v_c - v|$ as the diameter of $\mathcal{S}$.

Repeat

1. Divide each simplex $\mathcal{S}_i$ by using *bisection* method, which chooses the midpoint of one of the longest edges of the simplex $\mathcal{S}_i$, i.e., $v_m = \frac{1}{2}(v_r + v_s)$, where v_r and v_s are the end points of a longest edge of the simplex. In this case, the simplex $\mathcal{S}_i$ is subdivided into two simplices $\mathcal{S}_{i_1}$ and $\mathcal{S}_{i_2}$.

2. For each new simplex $\mathcal{S}_{i_j}$, compute the diameter $\delta(\mathcal{S}_{i_j})$ and $\mathbf{p}^*$ by solving (7.5) at $\mathbf{x}$ given by the center point of the simplex.

3. Find the current best solution to (7.3) and the maximal diameter δ^m in these new subdivided simplices.

Until $\delta^m < \epsilon$.

$d(\epsilon)$ decreases monotonically with the decreasing of ϵ, i.e., the solution given by Algorithm 7 monotonically converges to $\mathbf{x}^*$. As ϵ approaches zero, Algorithm 7 exhaustively searches every point in the simplex $\mathcal{S}$, thereby concluding the proof. □

Remarks: Algorithm 7 can be used to obtain an ϵ-optimal solution with $|\mathbf{x} - \mathbf{x}^*| \leq \epsilon$. That is to say, by controlling ϵ, one can strike a balance between the optimality and the computation time. Since finding the globally optimal solution requires centralized implementation, Algorithm 7 will be used only as a benchmark for performance evaluation of distributed algorithms.

2) DSPC algorithm

Next, we devise a distributed stochastic power control (DSPC) algorithm based on EDT [19] and SA [20]. Compared to the classical duality theory with nonzero duality gap for nonconvex optimization problems, EDT can guarantee zero duality gap between the primal and dual problems by utilizing nonlinear Lagrangian functions. This property allows for solving the nonconvex problem by its *extended dual* while preserving the global optimality in distributed implementation. To this end, we first introduce auxiliary variables and use EDT to transform (7.3) with the auxiliary variables into an unconstrained problem. Then, we solve the unconstrained problem by using the SA mechanism. Specifically, we define $t_l = \frac{U_l(\gamma_l(\mathbf{p}))}{x_l}$ and rewrite (7.3) as

$$\begin{aligned} \text{minimize} \quad & -\min_{l\in\mathcal{L}} t_l \\ \text{subject to} \quad & t_l x_l \leq U_l(\gamma_l(\mathbf{p})), \forall l \in \mathcal{L} \\ & \sum_{l\in\mathcal{L}} x_l = 1 \\ & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L} \\ & 0 \leq t_l, 0 \leq x_l \leq 1, \forall l \in \mathcal{L} \\ \text{variables} \quad & \{\mathbf{p}, \mathbf{x}, \mathbf{t}\}. \end{aligned} \tag{7.6}$$

Next, we use EDT to write the Lagrangian function for (7.6) as

$$L(\mathbf{p}, \mathbf{x}, \mathbf{t}, \alpha, \boldsymbol{\beta}) = -\min_{l\in\mathcal{L}} t_l + \alpha|\sum_{l\in\mathcal{L}} x_l - 1| + \sum_{l\in\mathcal{L}} \beta_l (t_l x_l - U_l(\gamma_l(\mathbf{p})))^+, \tag{7.7}$$

where $(y)^+ = \max(0, y)$, and $\alpha \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}^L$ are the penalty multipliers for penalizing the constraint violations. Based on EDT [19], there exist finite $\alpha^* \geq 0$ and $\beta_l^* \geq 0$, for all $l \in \mathcal{L}$, such that, for any $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$, $\forall\ l \in \mathcal{L}$, the solution to (7.7) is the same as (7.6). Note that (7.7) does not include the constraints of p_l, x_l, and t_l for each user, and there will be no constraint violation when each user updates these variables locally. Therefore, the minimization of (7.7) with respect to the primal variables ($\mathbf{p}$, $\mathbf{x}$, and $\mathbf{t}$) can be carried out individually by each user in a distributed fashion.

The next key step is to perform a stochastic local search by each user based on SA. Let t_l, x_l and p_l denote the primal values of the lth user, and t_l' and x_l' denote the new values randomly chosen by the lth user. Accordingly, $t_l' x_l'$ can be treated as a new target utility for the lth user. To achieve this target utility, the lth user updates p_l' by

$$p_l' = \min\left(\frac{U_l^{-1}(t_l' x_l')}{\gamma_l} p_l, P_l^{\max}\right), \tag{7.8}$$

where γ_l is the current SINR measured at the lth user's receiver. Note that (7.8) does not need any information of channel gains, except the feedback of SINR γ_l. Since (7.8) corresponds to the distributed power control algorithm of standard form as described in [151][7], it converges geometrically fast to the target utility. Thus, we assume that each user l updates p_l at a faster time-scale than t_l and x_l such that p_l always converges before the next update of t_l and x_l. Next, we use SA to update t_l and x_l in a stochastic operation. By using the analogy with annealing in metallurgy, SA was proposed in [20] to mimic the behavior of the microscopic constituents in heating and controlled cooling of a material. By allowing occasional uphill moves, SA is able to escape from

[7] A power control algorithm is of standard form, if the interference function (the effective interference each link must overcome) is positive, monotonic and scalable in power allocation [151].

the local optimal points. In particular, let Δ denote the difference between $L(p_l, x_l, t_l|\mathbf{p}_{-l}, \mathbf{x}_{-l}, \mathbf{t}_{-l}, \alpha, \boldsymbol{\beta})$ and $L(p'_l, x'_l, t'_l|\mathbf{p}'_{-l}, \mathbf{x}_{-l}, \mathbf{t}_{-l}, \alpha, \boldsymbol{\beta})$, where $\mathbf{y}_{-l}$ is the vector $\mathbf{y}$ without the lth user's variable. If $\Delta \geq 0$, i.e., t'_l, x'_l and p'_l reduce Lagrangian (7.7), then they are accepted with probability 1; otherwise, they are accepted with probability $\exp\left(\frac{\Delta}{T}\right)$, where T is a control parameter and sometimes it is called temperature. Note that, as T decreases, the acceptance of uphill move becomes less and less probable, and therefore a fine-grained search is needed. It has been shown that, as T approaches 0 according to a *logarithmic cooling schedule*, SA converges to a globally optimal point [18,144]. To compute Δ locally by each user l, user l needs to broadcast the terms t_l, x_l and $\beta_l(t_l x_l - U_l(\gamma_l(\mathbf{p})))^+$, whenever any of these terms changes.

Note that the target utility $t_l x_l$ may not be feasible, i.e., the target utility cannot be achieved even though the user transmits at the maximum power. In this case, it can be shown that the power of those users with feasible target utilities will converge to a feasible solution, whereas the other users that cannot achieve the target utility will continue to transmit at maximum power [7]. If some target utility is not feasible as T tends to 0, based on EDT, the current values of α and $\boldsymbol{\beta}$ do not satisfy $\alpha > \alpha^*$ or $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$. Therefore, each user l also needs to update α and β_l. In particular, if any constraint is violated, α and β_l are updated as follows:

$$\begin{aligned} \alpha &\leftarrow \alpha + \sigma|\sum_{l\in\mathcal{L}} x_l - 1|, \\ \beta_l &\leftarrow \beta_l + \varrho_l(t_l x_l - U_l(\gamma_l))^+, \ \forall l \in \mathcal{L}, \end{aligned} \tag{7.9}$$

where σ and ϱ_l are used to control the rate of updating α and β_l. A detailed description of DSPC algorithm is given in Algorithm 8.

Remarks:

1) In Algorithm 8, each user randomly picks $t_l' \in [U_l^{\max}, U_{tot}^{\max}]$, where $U_l^{\max}$ denotes the maximum utility of the user l when the user l transmits at the maximum power while the other users do not transmit, and $U_{tot}^{\max} = \sum_{l\in\mathcal{L}} U_l^{\max}$. Note that $U_l^{\max}$ can be computed by each user locally. Further, we assume that each user broadcasts $U_l^{\max}$ before running the algorithm so that $U_{tot}^{\max}$ is also known by each user.

2) In practice, after initialization, α and β_l increase in proportion to the violation of the corresponding constraint, which may lead to excessively large penalty values. Since it is beneficial to periodically scale down the penalty values to ease the unconstrained optimization, α and β_l are scaled down by multiplying with a random value (it can be chosen between 0.7 to 0.95 according to [19]), if the *penalty decrease condition* is satisfied, i.e., the maximum violation of constraints is not decreased after consecutively running Step 1 in Algorithm 8 several times, e.g., five times in [19].

3) In Algorithm 8, each user requires the knowledge of T and time epochs $\{\tau_1, \tau_2, ...\}$ to update t_l and x_l, which can be determined and informed to each user offline.

Proposition 7.2.2. *The distributed stochastic power control algorithm (Algorithm 8) converges almost surely to a globally optimal solution to (7.3), as temperature T in SA decreases to zero.*

Proof. To show that Algorithm 8 converges almost surely to a globally optimal solution to (7.3), we only need to show that when $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$, Algorithm 8 can converge almost surely to a globally optimal solution to (7.6), since (7.3) is equivalent to (7.6), and if the solution does not satisfy the constraints of (7.6), α and β_l will increase in proportion to the violation of

Algorithm 8 Distributed Stochastic Power Control (DSPC)

Initialization: Choose $\epsilon > 0$. Let $\alpha = 0$, $\beta_l = 0$, $\forall l \in \mathcal{L}$, and randomly choose $\mathbf{p}$, $\mathbf{x}$ and $\mathbf{t}$.

Step 1: update primal variables

Set $T = T_0$, and select a sequence of time epochs $\{\tau_1, \tau_2, ...\}$ in continuous time.

Repeat for each user l

1. Randomly pick $t_l' \in [U_l^{\max}, U_{tot}^{\max}]$ and $x_l' \in [0, 1]$, and update p_l' according to (7.8).
2. Keep sensing the change of $\beta_l(t_l x_l - U_l(\gamma_l(\mathbf{p})))^+$ broadcast by other users.
3. Compute Δ, and accept t_l', x_l', and p_l' with probability 1, if $\Delta \geq 0$, or with probability $\exp(\frac{\Delta}{T})$, otherwise.
4. Broadcast t_l' and x_l', if t_l' and x_l' are updated.
5. For each time epoch τ_i, update $T = T_0/\log(i+1)$.

Until $T < \epsilon$.

Step 2: update penalty variables

For each user l,

1. Update α and β_l according to (7.9), and scale down α and β_l, if the penalty decrease condition is satisfied.
2. Go to Step 1 until no constraint is violated.

---

the corresponding constraint. Since Algorithm 8 uses SA with the logarithmic cooling schedule, based on [18, 144] it can converge almost surely to a globally optimal solution to (7.7), which is also a globally optimal solution to (7.6) based on EDT when $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$ [19]. Hence Proposition 7.2.2 follows. □

Remarks: The DSPC algorithm can guarantee global optimality in a distributed manner without the need of channel information. In particular, it needs the information of t_l and x_l, and can adapt to channel variations by utilizing the SINR feedback. However, the convergence rate of DSPC is slow

due to the use of logarithmic cooling schedule.

3) Enhanced DSPC algorithm

It can be seen from Algorithm 8 that it is critical to find the optimal penalty variables α and $\boldsymbol{\beta}$ for computing (7.7). Moreover, a logarithmic cooling schedule is used to ensure convergence to a global optimum. To improve the convergence rate, we propose next an enhanced algorithm for DSPC (EDSPC) by empirically choosing the initial penalty values α_0 and $\boldsymbol{\beta}_0$ and employing a *geometric cooling schedule* [20], which reduces the temperature T in SA by $T = \xi T$, $0 < \xi < 1$, at each time epoch. Compared with the logarithmic cooling schedule, T converges to 0 much faster under the geometric cooling schedule, which in turn improves the convergence rate beyond DSPC. The resulting solution is given in Algorithm 9.

We note that although EDSPC converges much faster than DSPC, it may yield only near-optimal solutions. Based on EDT, we choose $\alpha_0 > \alpha^*$ and $\beta_{0l} > \beta_l^*$, $\forall\ l \in \boldsymbol{\mathcal{L}}$, to satisfy the optimality conditions for penalty variables. Obviously, by choosing large α_0 and β_{0l}, these conditions can be always satisfied. Nevertheless, very large penalties introduce heavy costs for constraint violations such that EDSPC may end up with a feasible but suboptimal solution. Therefore, the selection of initial penalty values plays a critical role in the performance of EDSPC and deserves more attention in future work. In practice, we can choose the initial penalties based on the maximum value of the constraint that is associated with each of the penalty variables. This choice performs well in the simulations. For example, we can choose $\beta_{0l} = U_l^{\max}$ for the constraint $t_l x_l \leq U_l(\gamma_l(\mathbf{p}))$.

Algorithm 9 Enhanced Distributed Stochastic Power Control (EDSPC)

Initialization: Choose $\epsilon > 0$. Let $\alpha = \alpha_0$, $\beta_l = \beta_{0l}$, $\forall l \in \mathcal{L}$, and randomly choose $\mathbf{p}$, $\mathbf{x}$ and $\mathbf{t}$.
Set $T = T_0$, and select a sequence of time epochs $\{\tau_1, \tau_2, ...\}$ in continuous time.
Repeat for each user l

1. Randomly pick $t_l' \in [U_l^{\max}, U_{tot}^{\max}]$ and $x_l' \in [0, 1]$, and update p_l' according to (7.8).
2. Keep sensing the change of $\beta_l(t_l x_l - U_l(\gamma_l(\mathbf{p})))^+$ broadcast by other users.
3. Compute Δ, and accept t_l', x_l', and p_l' with probability 1, if $\Delta \geq 0$, or with probability $\exp(\frac{\Delta}{T})$, otherwise.
4. Broadcast t_l' and x_l', if t_l' and x_l' are updated.
5. For each time epoch τ_i, update $T = \xi T$.

Until $T < \epsilon$.

7.2.5 Performance Evaluation

In this section, we evaluate the utility and convergence performance of Algorithms 8 and 9 (DSPC[8] and EDSPC). We consider a wireless network with six links randomly distributed on a 10m-by-10m square area. The channel gains h_{lk} are equal to d_{lk}^{-4}, where d_{lk} represents the distance between the transmitter of user l and the receiver of user k. We assume $U_l(\gamma_l(\mathbf{p})) = \log(1 + \gamma_l(\mathbf{p}))$, $P_l^{\max} = 1$ and $n_l = 10^{-4}$ for all $l \in \mathcal{L}$, and consider one randomly generated

[8]The geometric cooling schedule is employed to accelerate the convergence rate of DSPC in the simulation. DSPC updates penalty values until they satisfy the threshold-based optimality condition.

realization of channel gains given by

$$\mathbf{H} = \begin{bmatrix} 0.3318 & 0.0049 & 0.0141 & 0.0021 & 0.0016 & 0.0007 \\ 0.0031 & 0.9554 & 0.0063 & 0.0140 & 0.0012 & 0.0025 \\ 0.0155 & 0.0042 & 0.6166 & 0.0046 & 0.0108 & 0.0018 \\ 0.0017 & 0.2188 & 0.0340 & 0.6754 & 0.0062 & 0.0215 \\ 0.0020 & 0.0017 & 0.2216 & 0.0042 & 0.2955 & 0.0028 \\ 0.0007 & 0.0079 & 0.0254 & 0.2553 & 0.0404 & 0.3025 \end{bmatrix}.$$

Fig. 7.5 shows how the total utility in the EDSPC algorithm converges over time, when we choose all the initial penalty values equal to 10. Also, we choose $\xi = 0.9$, $\rho = 1$ and $\varrho = 1$, and use Algorithm 7 as a benchmark to evaluate the optimal performance. As shown in Fig. 7.5, the EDSPC algorithm approaches the optimal utility, when the initial penalty values are carefully chosen. Moreover, the convergence rate of the EDSPC algorithm is much faster than DSPC, since DSPC continues updating the penalty values even after the optimal solution is found for the current penalty values. Fig. 7.6 illustrates the average performance (with confidence interval) of DSPC, EDSPC, and SEER under 100 random initializations, with the same system parameters as used in Fig. 7.5. As shown in Fig. 7.6, both DSPC and EDSPC are robust against the variations of initial values.

Fig. 7.5 and 7.6 compare the proposed algorithms with the SEER and ADP. As mentioned in Section 7.1, ADP can only guarantee local optimality. Therefore, for nonconvex problems (e.g., in this example), ADP may converge to a suboptimal solution. As noted in [17], the performance of SEER heavily hinges on the control parameter that can be challenging to choose in online operation. In contrast, DSPC can approach the globally optimal solution regardless of the initial parameter selection, but the convergence rate may be

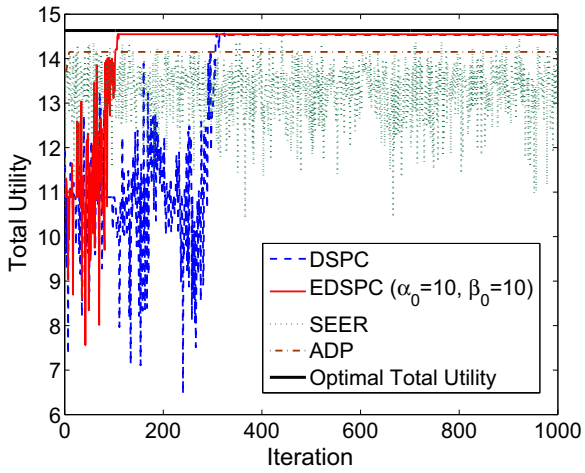

Figure 7.5: Convergence performance of DSPC, EDSPC, SEER and ADP.

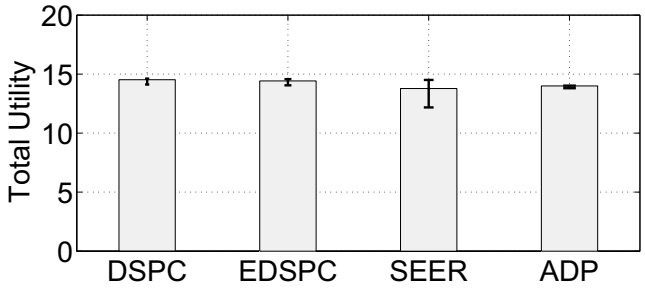

Figure 7.6: Comparison of the average utility performance (with confidence interval) of DSPC, EDSPC, SEER and ADP.

slower. Furthermore, EDSPC improves the convergence rate, but in this case the initial penalty values would impact how close it can approach the optimal point. In terms of message passing, our algorithms do not require individual links to know the channel gains (including its own channel gain), the receiver SINR of the other links and the signal power of the other links, which are all

needed in the SEER algorithm.

7.3 Joint Scheduling and Power Control for Stability of Queueing Systems

In Section 7.2, we studied the distributed power allocation, by using DSPC and EDSPC, for utility maximization in the saturated case with uninterrupted packet traffic. In this section, we generalize the study by considering a queueing system with dynamic packet arrivals and departures. Specifically, we develop a joint scheduling and power allocation policy to stabilize packet queues by integrating our power control algorithms with the celebrated back-pressure algorithm [145].

7.3.1 Stability Region and Throughput Optimal Power Allocation Policy

Consider the same wireless network model with L links as in Section 7.2. We assume that there are S classes of users in the system, and that the traffic brought by users of class s follows $\{A_{sl}(t)\}_{t=1}^{\infty}$, which are *i.i.d.* sequences of random variables for all $l = 1, ..., L$ and $s = 1, ..., S$, where $A_{sl}(t)$ denotes the amount of traffic generated by users of class s that enters the link l in slot t. We assume that the second moments of the arrival process $\{A_{sl}(t)\}_{t=1}^{\infty}$ are finite. Let $Q_{T(l)}^{s}(t)$ and $Q_{R(l)}^{s}(t)$ denote the current backlog in the queue of class s in slot t on the transmitter and receiver sides of link l, respectively. The queue length $Q_{T(l)}^{s}(t)$ evolves over time as

$$Q_{T(l)}^{s}(t+1) = \max(Q_{T(l)}^{s}(t) - r_{l}^{s}(t), 0) + A_{sl}(t) + \sum_{\{m|T(l)=R(m), m\in\mathcal{L}\}} r_{m}^{s}(t), \tag{7.10}$$

where $r_{l}^{s}(t)$ denotes the transmission rate of link l for users of class s. The third term in (7.10) denotes the traffic from the other links. The queue length process $\{Q_{T(l)}^{s}(t)\}_{t=1}^{\infty}$ forms a Markov chain.

Let ψ_s denote the first moment of $\{A_{sl}(t)\}_{t=1}^{\infty}$, i.e., the load brought by users of class s. As is standard [145, 146, 152], the stability region is defined as follows.

Definition 7.3.1. *The stability region Λ is the closure of the set of all $\{\psi_s\}_{s=1}^{S}$ for which there exists some feasible power allocation policy under which the system is stable, i.e., $\Lambda = \bigcup_{\mathbf{p}\in\mathcal{P}} \Lambda(\mathbf{p})$, where $\Lambda(\mathbf{p}) = \{\{\psi_s\}_{s=1}^{S} | \sum_{s=1}^{S} E_{sl}\psi_s < r_l(\mathbf{p}), \forall l\}$, $\mathcal{P}$ denotes the set of feasible power allocation, $r_l(\mathbf{p})$ denotes the rate of link l under power allocation $\mathbf{p}$, and $E_{sl} = 1$ is the indicator that the path of users of class s uses link l, and $E_{sl} = 0$, otherwise.*

For the sake of comparison, the throughput region[9] $\mathcal{F}$ of the corresponding saturated case is defined as the set of all feasible link rates, i.e., $\mathcal{F} = \{\mathbf{r} | r_l = r_l(\mathbf{p}), \mathbf{p} \in \mathcal{P}\}$. In general, the throughput region $\mathcal{F}$ may be different from the stability region Λ, except for some special cases (e.g., in slotted ALOHA systems the throughput region and the stability region are the same [153] for two links and in a multiple-access channel the information theoretic capacity region is equivalent to its stability region under specific feedback assumptions [154]).

The queueing system is stable if the arrival rates of packet queues are less than the service rates such that the queue lengths do not grow to infinity [155]. In order to stabilize packet queues, it is critical to find the optimal scheduling and power allocation policy that maximizes the weighted sum rate given by (7.11). By integrating our power control algorithms and the back-pressure algorithm, we propose a joint scheduling and power allocation policy presented in Algorithm 10 to stabilize the queueing system.

[9]Note that the feasible utility region $\mathcal{F}$ defined in Section 7.2 is the throughput region, when the utility function is the same as the rate function.

Proposition 7.3.1. *The joint scheduling and power allocation policy (Algorithm 10) can stabilize the system such that* $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{l,s} E\{Q_l^s(\tau)\} < \infty$*, when the traffic load* $\{\psi_s\}_{s=1}^S$ *is strictly interior to the stability region* Λ*, i.e., there exists some* $\epsilon > 0$ *such that* $\{\psi_s + \epsilon\}_{s=1}^S \in \Lambda$.

The proof is similar to that in [43, 146], and is omitted for brevity.

Note that Algorithm 10 can be viewed as a dynamic back-pressure and resource allocation policy [43], crafted towards solving the weighted sum rate maximization problem (7.11). Specifically, by using the DSPC algorithm, Algorithm 10 can be implemented distributively to find the globally optimal resource allocation. We should caution that EDSPC can be applied to improve the convergence rate of Stage 2 in Algorithm 10 but it may render a suboptimal schedule (i.e., it can not stabilize all possible $\{\psi_s\}_{s=1}^S$ within Λ), due to the fact that EDSPC may not always find the global optimal power allocation.

To reduce the complexity, we can consider a policy that computes (7.11) periodically every few slots, and it can be shown that this policy can also stabilize the system, when $\{\psi_s\}_{s=1}^S$ is strictly interior to the stability region Λ [156, 157].

7.3.2 Performance Evaluation

In this section, we present numerical results to illustrate the use of Algorithm 10 for stabilizing a queueing system. We consider a one-hop network (i.e., $\mathbf{E} = \{E_{sl}\}$ is the identity matrix) with two users (classes), where the channel gains are $h_{11} = 0.3$, $h_{12} = 0.5$, $h_{21} = 0.03$, and $h_{22} = 0.8$, and the noise power is 0.1 for each link. The maximum transmission power is set to 1 and 2 for links 1 and 2, respectively. Besides, we assume that the users of class s arrive at the network according to a Poisson process with rate λ_s, and that the size

Algorithm 10 Joint Scheduling and Power Allocation Policy

Stage 1: For each link l, select a link weight according to $w_l(t) = \max_{s=1,\ldots,S} D_l^s(t)$, where the difference of queue lengths of class s is $D_l^s(t) = \max(Q_{T(l)}^s(t) - Q_{R(l)}^s(t), 0)$, if the receiver of link l is not the destination of class s's traffic, and $D_l^s(t) = Q_{T(l)}^s$, otherwise.

Stage 2: Compute the optimal power allocation $\mathbf{p}^*$ in each slot t by solving the following problem with DSPC algorithm

$$\mathbf{p}^* = \arg\max_{\mathbf{p}} \sum_{l=1}^{L} w_l(t) r_l(\mathbf{p}). \tag{7.11}$$

Thus, the transmission rate of link l in slot t is given by $r_l(\mathbf{p}^*) = \log(1 + \gamma_l(\mathbf{p}^*))$.

Stage 3: Let $s_l^* = \arg\max_{s=1,\ldots,S} D_{sl}(t)$ denote the class scheduled in slot t; if multiple classes satisfy this condition, then s_l^* is randomly chosen as one of these classes. Then, schedule these classes according to the solution given by Stage 2.

of packet batch for users of class s follows an exponential distribution with mean ν_s. The load brought by users of class s is then $\psi_s = \lambda_s \nu_s$. Fig. 7.7 shows the stability region Λ and compares it with the throughput region $\mathcal{F}$ of the corresponding saturated case. The stability region follows from the union of link rates that are conditioned on whether the other link is backlogged or not [153, 154]. First, we derive the stability region for the given power allocation. Then, we vary power allocation in the feasible region, and by taking the envelope of these regions, we obtain the overall stability region shown in Fig. 7.7. However, different from the previous cases, where the throughput region is the same as the stability region, e.g., in a slotted ALOHA system with two links [153] and in a multiple-access channel [154], the throughput region $\mathcal{F}$ under the SINR model is strictly smaller than the stability region (due to the underlying nonconvex optimization problem), as observed from Fig. 7.7, which is the convex hull of $\mathcal{F}$, i.e., $Co(\mathcal{F})$, achievable by timesharing

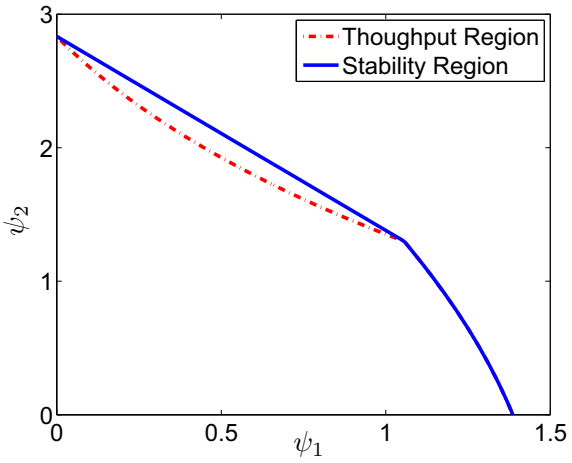

Figure 7.7: Comparison of the stability region and the throughput region.

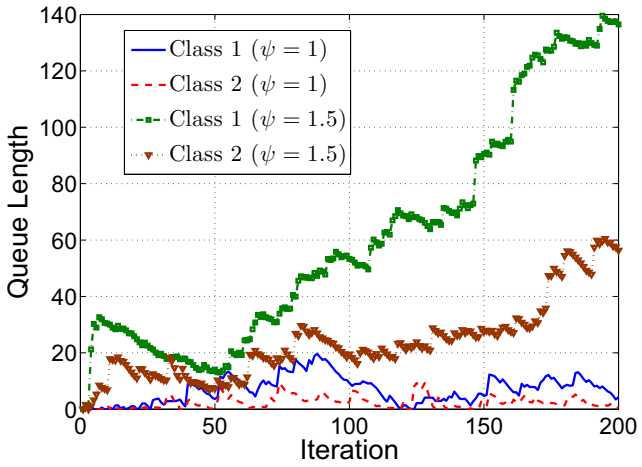

Figure 7.8: Comparison of sample paths of a user's queue length for different traffic loads.

across different transmission modes[10].

Then, we vary the arrival rate λ and the average batch size ν to change the traffic intensity $\psi = \lambda\nu$. Assuming that the arrival rate and the average batch size of each user are the same, we compare in Fig. 7.8 the sample paths

[10]The transmission mode is defined as the transmission rate pair within the throughput region $\mathcal{F}$.

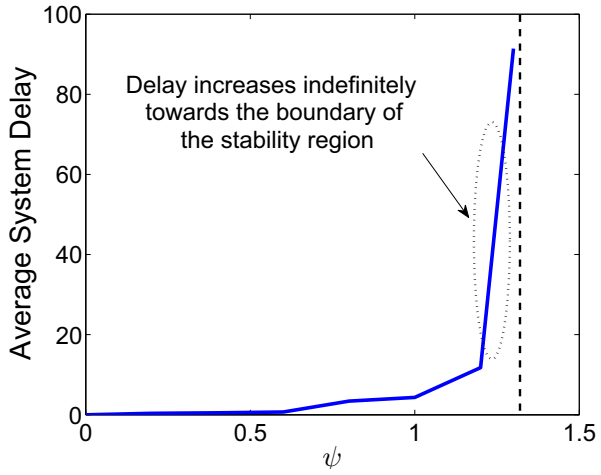

Figure 7.9: Average delay of the system vs. system loads.

of each user's queue length for $\psi = 1$ ($\lambda = 1$, $\nu = 1$) with $\psi = 1.5$ ($\lambda = 1.5$, $\nu = 1$). When $\psi = 1$, which falls in the stability region shown in Fig. 7.7, the system is stabilized by using Algorithm 10. On the other hand, the system becomes unstable when $\psi = 1.5$, which is outside the stability region. Fig. 7.9 illustrates the average delay of the system as a function of the arrival rates. The delay is finite for small loads and grows unbounded when the loads are outside the stability region.

7.4 Power Control for Multicast Communications

Due to wireless multicast advantage [148], multicasting enables efficient data delivery to multiple recipients with a single transmission. In this section, we extend the distributed stochastic power control algorithms in Section 7.2 to support multicast communications.

7.4.1 System Model

Beyond the model described in Section 7.2, we consider that each user l has one transmitter and a set $\mathcal{M}_l$ of receivers. The corresponding transmission rate, r_l, is determined by the bottleneck link among these transmitter-receiver pairs, i.e., $r_l = \min_{m \in \mathcal{M}_l} r_{lm}$, where r_{lm} denotes the link rate between the transmitter of user l and its receiver m, and it is calculated from the Shannon rate $\log(1 + \gamma_{lm}(\mathbf{p}))$ for Gaussian, flat fading channels. Here, we do not consider the general broadcast capacity region but rather focus on maximizing the bottleneck link rates.

7.4.2 Network Utility Maximization

We seek to find the optimal power allocation $\mathbf{p}^*$ that maximizes the overall system utility subject to the power constraints in multicast communications, as follows:

$$\begin{aligned} \text{maximize} \quad & \sum_{l \in \mathcal{L}} U_l(r_l) \\ \text{subject to} \quad & r_l = \min_{m \in \mathcal{M}_l} r_{lm}, \forall\ l \in \mathcal{L} \\ & r_{lm} = \log(1 + \gamma_{lm}(\mathbf{p})), \forall\ l \in \mathcal{L}, m \in \mathcal{M}_l \\ & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} \quad & \{\mathbf{p}, \{r_l\}, \{r_{lm}\}\}. \end{aligned} \tag{7.12}$$

Similar to (7.2), (7.12) is nonconvex due to the complicated interference coupling between individual links. Different from the techniques used in Section 7.2, we relax $r_l = \min_{m \in \mathcal{M}_l} r_{lm}$ in (7.12) as $r_l \leq \log(1 + \gamma_{lm}(\mathbf{p})), \forall\ l \in \mathcal{L}, m \in \mathcal{M}_l$, in order to devise distributed algorithms solving (7.12). Thus,

(7.12) can be rewritten as

$$\begin{aligned} \text{maximize} \quad & \sum_{l\in\mathcal{L}} U_l(r_l) \\ \text{subject to} \quad & r_l \le \log(1+\gamma_{lm}(\mathbf{p})), \forall\ l \in \mathcal{L}, m \in \mathcal{M}_l \\ & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} \quad & \{\mathbf{p}, \mathbf{r}\}. \end{aligned} \tag{7.13}$$

7.4.3 Distributed Global Optimization Algorithms

We develop next distributed algorithms that can find the globally optimal solutions to (7.13) based on EDT and SA. To this end, we first rewrite the optimization problem (7.13) as

$$\begin{aligned} \text{minimize} \quad & -\sum_{l\in\mathcal{L}} U_l(r_l) \\ \text{subject to} \quad & r_l \le \log(1+\gamma_{lm}(\mathbf{p})), \forall\ l \in \mathcal{L}, m \in \mathcal{M}_l \\ & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} \quad & \{\mathbf{p}, \mathbf{r}\}. \end{aligned} \tag{7.14}$$

Next, we use EDT to write the Lagrangian function for (7.14) as

$$L(\mathbf{p}, \mathbf{r}, \{\alpha_{lm}\}) = -\sum_{l\in\mathcal{L}} U_l(r_l) + \sum_{l\in\mathcal{L}, m\in\mathcal{M}_l} \alpha_{lm}(r_l - \log(1+\gamma_{lm}(\mathbf{p})))^+, \tag{7.15}$$

where $\alpha_{lm} \in \mathbb{R}$ are the penalty multipliers. From EDT, there exist finite $\alpha_{lm}^* \ge 0$ for all $l \in \mathcal{L}, m \in \mathcal{M}_l$ such that, for any $\alpha_{lm} > \alpha_{lm}^*$, $\forall\ l \in \mathcal{L}, m \in \mathcal{M}_l$, the solution to (7.15) is the same as (7.14) [19]. Note that (7.15) does not include the constraint of p_l for each user. Therefore, there will be no constraint violation when each user updates the transmission power locally, while minimizing (7.15) in a distributed operation.

As in Section 7.2, the key step is to let each user perform a local stochastic search based on SA. Let r_l and p_l denote the primal values of the lth user, and r_l' denote the new value randomly chosen by the lth user, which

is treated as a new target transmission rate for the lth user. Different from the unicast communications case, the lth user updates p_l' by

$$p_l' = \min\left(\frac{e^{r_l'}-1}{\min\limits_{m\in\mathcal{M}_l}\gamma_{lm}}p_l, P_l^{\max}\right), \tag{7.16}$$

where γ_{lm} is the current SINR measured at the receiver m of user l. Note that (7.16) does not need any information of the channel gains, except the feedback of SINR γ_{lm} from the intended receivers. Since (7.16) is in standard form as described in [151], it converges geometrically fast to the target transmission rate. The steps to update r_l and α_{lm} are similar to DSPC Algorithm 8 in Section 7.2. Note that the target transmission rate r_l may not be feasible, i.e., the target utility cannot be achieved even though the user transmits at the maximum power. In this case, it can be shown that the power of those users with feasible target transmission rates will converge to a feasible solution, whereas the other users that cannot achieve the target transmission rate will continue to transmit at maximum power [7]. A detailed description of DSPC algorithm for multicast communications is presented in Algorithm 11.

Remarks: In Algorithm 11, each user randomly picks $r_l' \in [0, r_l^{\max}]$, where $r_l^{\max} = \min\limits_{m\in\mathcal{M}_l} r_{lm}^{\max}$, and $r_{lm}^{\max}$ is the maximum link rate between the transmitter of the user l and its receiver m, when the user l transmits at the maximum power while the other users do not transmit.

Proposition 7.4.1. *The distributed stochastic power control algorithm for multicast communications (Algorithm 11) converges almost surely to a globally optimal solution to (7.13), as temperature T in SA approaches zero.*

Proof. The proof is based on EDT and SA arguments, and follows similar steps used in the proof of Proposition 7.2.2, and it is omitted here for brevity. □

To improve the convergence rate, we also propose an enhanced algorithm for Algorithm 11 by empirically choosing the initial penalty values and employing a geometric cooling schedule. The resulting algorithm is given in Algorithm 12. Similar to the unicast case, Algorithms 11 and 12 do not need any knowledge of channel information (or the bottleneck link) and they are dynamically updated by the SINR feedback from the intended receivers.

Algorithm 11 DSPC for Multicast Communications

Initialization: Choose $\epsilon > 0$. Let $\alpha_{lm} = 0, \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l$ and randomly choose $\mathbf{r}$ and $\mathbf{p}$.

Step 1: update primal variables

Set $T = T_0$, and select a sequence of time epochs $\{\tau_1, \tau_2, ...\}$ in continuous time.

Repeat for each user l

1. Randomly pick $r_l' \in [0, r_l^{\max}]$, and update p_l' according to (7.16).
2. Keep sensing the change of $\sum_{m \in \mathcal{M}_l} \alpha_{lm}(r_l - \log(1+\gamma_{lm}(\mathbf{p})))^+$ broadcast by other users.
3. Let Δ be the difference between $L(\mathbf{p}, r_l|\mathbf{r}_{-l}, \{\alpha_{lm}\})$ and $L(\mathbf{p}', r_l'|\mathbf{r}_{-l}, \{\alpha_{lm}\})$, and accept r_l' and p_l' with probability 1, if $\Delta \geq 0$, or with probability $\exp(\frac{\Delta}{T})$, otherwise.
4. Broadcast $U_l(r_l')$, if r_l' is accepted.
5. For each time epoch τ_i, update $T = T_0/\log(i+1)$.

Until $T < \epsilon$.

Step 2: update penalty variables

For each user l,

1. Update $\alpha_{lm} \leftarrow \alpha_{lm} + \varrho_{lm}(r_l - \log(1+\gamma_{lm}(\mathbf{p})))^+$, and scale down α_{lm}, if the condition of penalty decrease is satisfied.
2. Go to Step 1 until no constraint is violated.

7.4.4 Performance Evaluation

In this section, we evaluate the performance of Algorithms 11 and 12 for multicast communications. We consider a wireless network with four transmitters

Algorithm 12 EDSPC for Multicast Communications

Initialization: Choose $\epsilon > 0$. Let $\alpha_{lm} = \alpha_{lm}^0$, $\forall\ l \in \mathcal{L}, m \in \mathcal{M}_l$ and randomly choose $\mathbf{r}$ and $\mathbf{p}$.
Set $T = T_0$, and select a sequence of time epochs $\{\tau_1, \tau_2, ...\}$ in continuous time.
Repeat for each user l

1. Randomly pick $r_l' \in [0, r_l^{\max}]$, and update p_l' according to (7.16).
2. Keep sensing the change of $\sum_{m \in \mathcal{M}_l} \alpha_{lm}(r_l - \log(1+\gamma_{lm}(\mathbf{p})))^+$ broadcast by other users.
3. Let Δ be the difference between $L(\mathbf{p}, r_l|\mathbf{r}_{-l}, \{\alpha_{lm}\})$ and $L(\mathbf{p}', r_l'|\mathbf{r}_{-l}, \{\alpha_{lm}\})$, and accept r_l' and p_l' with probability 1, if $\Delta \geq 0$, or with probability $\exp(\frac{\Delta}{T})$, otherwise.
4. Broadcast $U_l(r_l')$, if r_l' is accepted.
5. For each time epoch τ_i, update $T = \xi T$.

Until $T < \epsilon$.

and each transmitter has two receivers. These transmitters and receivers are randomly placed on a 10m-by-10m square area. The channel gains h_{lm} are equal to d_{lm}^{-4}, where d_{lm} represents the distance between the transmitter l and the receiver m. The channel gains h_{lm} are equal to d_{lm}^{-4}, where d_{lm} represents the distance between the transmitter l and the receiver m. We assume $U_l(r_l) = r_l$, $P_l^{\max} = 1$, and $n_{lm} = 10^{-4}$ for all $l \in \mathcal{L}$ and $m \in \mathcal{M}_l$. Fig. 7.10 illustrates the fast convergence performance of Algorithms 11 and 12 in multicast communications.[11] Besides, we examine the average performance (with confidence interval) of DSPC and EDSPC for multicast communications under 100 random initializations with the same system parameters as in the case shown in Fig. 7.10. As illustrated in Fig. 7.11, both algorithms 11 and 12 are robust against the initial value variations.

[11]The other existing algorithms have been specifically designed for unicast communications; therefore, they are excluded here from the performance comparison.

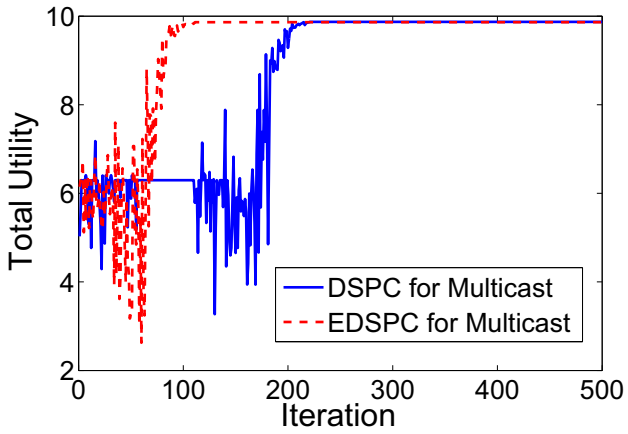

Figure 7.10: Convergence performance of DSPC and EDSPC for multicast communications, where $\alpha_{lm}^0 = 20, \forall\ l \in \mathcal{L}, m \in \mathcal{M}_l$.

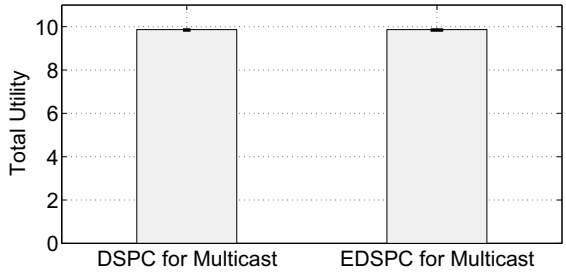

Figure 7.11: Comparison of average performance (with confidence interval) of DSPC and EDSPC for multicast.

7.5 Conclusion

We studied the distributed power control problem of optimizing the system utility as a function of the achievable rates in wireless ad hoc networks. Based on the observation that the global optimum lies on the boundary of the feasible region for unicast communications, we focused on the equivalent but more structured problem in the form of maximizing the minimum weighted utility. Appealing to extended duality theory, we decomposed the minimum weighted

utility maximization problem into subproblems by using penalty multipliers for constraint violations. We then proposed a distributed stochastic power control (DSPC) algorithm to seek a globally optimal solution, where each user stochastically announces its target utility to improve the total system utility via simulated annealing. In spite of the nonconvexity of the underlying problem, the DSPC algorithm can guarantee global optimality, but only with a slow convergence rate. Therefore, we proposed an enhanced distributed algorithm (EDSPC) to improve the convergence rate with geometric cooling schedule in simulated annealing. We then compared DSPC and EDSPC with the existing power control algorithms and verified the optimality and complexity reduction.

Next, we proposed the joint scheduling and power allocation policy for queueing systems by integrating our distributed power control algorithms with the back-pressure algorithm. The stability region was evaluated, which is shown to be strictly greater than the throughput region in the corresponding saturated case. Beyond unicast communications, we generalized our power control algorithms to multicast communications by jointly maximizing the minimum rates on bottleneck links in different multicast groups. Our distributed stochastic power control approach guarantees global optimality without the need of channel information, while reducing the computation complexity, in general systems with unicast and multicast communications, and applies to both backlogged and random traffic patterns.

Chapter 8

CONCLUSION AND DISCUSSION

In this dissertation, we have studied stochastic optimization and real-time scheduling in cyber-physical systems, with the focus on real-time wireless systems, energy systems and distributed control systems. The main body of this dissertation can be categorized into three parts. The first part (Chapters 2-3) focuses on the scheduling problems in the real-time wireless systems by using dynamic programming. The second part (Chapters 4-5) targets at the EV charging and the wind integration in the energy systems. The third part (Chapters 6-7) studies the distributed optimization of network resource allocation in cognitive radio networks and wireless ad-hoc networks respectively. In what follows, we summarize our key contributions and outline possible future directions.

8.1 Real-time Wireless Systems

In Chapter 2, we considered the problem of adaptive NC for multimedia traffic with hard deadlines over a single-hop wireless network. We formulated the sequential block size adaptation problem for NC as a Markov decision process. By exploring the structural properties of the problem, we developed the MBIA that can solve for the optimal block size adaptation in polynomial time. Besides, we developed the joint real-time scheduling and channel learning scheme with adaptive NC that can adapt to wireless channel dynamics in case the perfect channel information is not available at the scheduler. Then, we generalized the study of adaptive NC to multiple flows with hard deadlines

and long-term delivery ratio constraints, and developed a low-complexity online scheduling algorithm integrated with the MBIA. Finally, we performed high fidelity wireless emulation tests with real radio transmissions to demonstrate the feasibility of the MBIA in finding the optimal block size in real time. It is of great interest to study unicast communications of multiple flows with NC across different flows and investigate the corresponding NC schemes with deadline constraints for real-time traffic scheduling.

In Chapter 3, we explored the optimization of joint channel learning and scheduling of real-time traffic with hard deadlines and packet-level priorities, under Markovian channels. Formulating the scheduling problem as a partially observable Markov decision process, we identified the unique two-timescale nature of the "exploitation vs exploration" tradeoffs, where idling provides a new dimension to the action space. Via a case study, we showed that idling is an optimal action under certain system states in the light traffic regime, further underlining the fundamental tradeoffs between exploitation and exploration/idling. In contrast, for the special case with a symmetric two-user system, we showed that the scheduling problem exhibits a unique structure – the equivalence with the genie-aided system – that renders the *non-idling* greedy policy optimal. A natural next step is to study if the greedy policy is optimal in the symmetric system with more than two users. In the light traffic regime, we discovered that idling is an optimal action under certain system states. Obviously, when the number of packets generated in each period is greater than the number of slots of a period, idling cannot be an optimal action. It is interesting to quantify this light traffic regime, where idling can be an optimal action in heterogeneous systems. Another interesting direction is to study the optimal policy in the heavy traffic regime and the performance

of the corresponding greedy policy.

8.2 Energy Systems

In Chapter 4, we studied risk-aware day-ahead scheduling and real-time dispatch for plug-in electric vehicles. We proposed the risk-aware day-ahead scheduling algorithm that minimizes the EV charging cost and the risk of the load mismatch between the forecasted and the actual EV loads, due to the random driving activities of EVs. Although the risk-aware day-ahead scheduling problem is nonconvex, by utilizing the *hidden convexity* structure, it can be recast as a two-stage stochastic linear program, which can be solved by using the L-shaped method. Further, the distributed risk-aware real-time dispatch algorithm was developed, where the aggregator only needs to compute the shadow prices for each EV to optimize its own charging strategy in a distributed manner. Based on real data, the simulation results showed that the proposed risk-aware day-ahead scheduling algorithm can reduce not only the overall charging cost, but also the peak demand of EV charging. It is of great interest to study the EV charging strategy that takes into account the random EV arrivals and departures in real time.

In Chapter 5, we investigated joint optimization of ED and interruptible load management by using short-term wind farm generation forecast. Specifically, a finite state Markov chain model for wind farm generation forecast was developed based on spatial and temporal characterizations of wind turbine power outputs. The proposed joint optimization of ED and interruptible load management was cast as a Markov decision process based dynamic programming problem. To reduce the complexity of this joint optimization problem, the greedy policy was used. Further, by leveraging the convexity properties,

the proposed ED problem was transformed into a stochastic program by using the point forecast of wind generation. Numerical studies, via the IEEE Reliability Test System – 1996 and realistic wind measurement data from an actual wind farm, demonstrated the significant benefits obtained by integrating the Markov-chain-based forecast and the interruptible load management, compared with conventional wind-speed-based forecasting methods. It is of great interest to develop systematic approaches to address the uncertainty of wind generation in electricity markets in a cost-effective manner. For example, secondary ancillary services markets can be considered in the proposed ED framework.

8.3 Distributed Control Systems

In Chapter 6, we studied pricing-based distributed spectrum access in cognitive radio networks, where SUs compete via random access for available spectrum opportunities. We developed two models: one with the monopoly PU market and the other with the multiple PU market. For the monopoly PU market model, we applied the revenue maximization approach to characterize the appropriate choice of flat and usage prices, and derived a Pareto optimal solution, which was shown to be near-optimal. More importantly, this Pareto optimal solution exhibits a decentralized structure, i.e., the Pareto optimal pricing strategy and access probabilities can be computed by the PU and the SUs locally. We also analyzed a PU profit maximization problem by examining the tradeoff between the PU's utility and its revenue.

We then studied the multiple PU market model by casting the competition among PUs as a three-stage Stackelberg game, in terms of access prices and the offered spectrum opportunities. We showed the existence of the Nash

equilibrium, and derived a necessary and sufficient condition for the uniqueness of Nash equilibrium for the case with homogeneous PUs. Intuitively, this condition can be used by PUs to decide whether to join in the competition or not, i.e., when the number of PUs grows larger than the threshold, the competition among PUs is too strong, indicating that it is unprofitable for a PU to sell spectrum to SUs. Then we developed an iterative algorithm for strategy adaption to achieve the Nash equilibrium. It remains open to characterize the condition for the uniqueness of Nash equilibrium for the case with heterogeneous PUs. Another interesting direction is to investigate transient behaviors corresponding to dynamic spectrum access, in the presence of spectrum hole dynamics.

In Chapter 7, we studied the distributed power control problem of optimizing the system utility as a function of the achievable rates in wireless ad hoc networks. Based on the observation that the global optimum lies on the boundary of the feasible region for unicast communications, we focused on the equivalent but more structured problem in the form of maximizing the minimum weighted utility. Appealing to extended duality theory, we decomposed the minimum weighted utility maximization problem into subproblems by using penalty multipliers for constraint violations. We then proposed a distributed stochastic power control (DSPC) algorithm to seek a globally optimal solution, where each user stochastically announces its target utility to improve the total system utility via simulated annealing. In spite of the nonconvexity of the underlying problem, the DSPC algorithm can guarantee global optimality, but only with a slow convergence rate. Therefore, we proposed an enhanced distributed algorithm (EDSPC) to improve the convergence rate with geometric cooling schedule in simulated annealing. We then compared DSPC and

EDSPC with the existing power control algorithms and verified the optimality and complexity reduction.

Next, we proposed the joint scheduling and power allocation policy for queueing systems by integrating our distributed power control algorithms with the back-pressure algorithm. The stability region was evaluated, which is shown to be strictly greater than the throughput region in the corresponding saturated case. Beyond unicast communications, we generalized our power control algorithms to multicast communications by jointly maximizing the minimum rates on bottleneck links in different multicast groups. Our distributed stochastic power control approach guarantees global optimality without the need of channel information, while reducing the computation complexity, in general systems with unicast and multicast communications, and applies to both backlogged and random traffic patterns.

REFERENCES

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proc. Design Autom. Conf.*, June 2010, pp. 731–736.

[2] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[3] K. Cory and B. Swezey, "Renewable portfolio standards in the states: balancing goals and implementation strategies," NREL Technical Report TP-670-41409, Dec. 2007.

[4] L. Xie, P. Carvalho, L. Ferreira, J. Liu, B. Krogh, N. Popli, and M. Ilic, "Wind integration in power systems: operational challenges and possible solutions," *Proc. IEEE*, vol. 99, no. 1, pp. 214–232, 2011.

[5] K. Porter and J. Rogers, "Survey of variable generation forecasting in the west," NREL Subcontract Report SR-5500-54457, Apr. 2012.

[6] S. Huang, X. Liu, and Z. Ding, "Optimal transmission strategies for dynamic spectrum access in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 12, pp. 1636–1648, 2009.

[7] M. Chiang, P. Hande, T. Lan, and C. W. Tan, "Power control in wireless cellular networks," *Foundations and Trends in Networking*, vol. 2, no. 4, pp. 381–553, 2008.

[8] D. Julian, M. Chiang, D. O. Neill, and S. Boyd, "QoS and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," in *Proc. IEEE INFOCOM*, June 2002, pp. 477–486.

[9] M. Chiang, C. W. Tan, D. Palomar, D. O. Neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 1, no. 7, pp. 2640–2651, 2007.

[10] M. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power control scheme in wireless cellular systems," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 210–221, 2003.

[11] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: complexity and duality," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 57–73, 2008.

[12] P. Hande, S. Rangan, M. Chiang, and X. Wu, "Distributed uplink power control for optimal SIR assignment in celluar data networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1420–1433, 2008.

[13] J. Huang, R. Berry, and M. Honig, "Distributed interference compensation for wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1074–1084, 2006.

[14] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291–303, 2002.

[15] T. Alpcan, T. Basar, R. Srikant, and E. Altman, "CDMA uplink power control as a noncooperative game," *Wireless Networks*, vol. 8, no. 6, pp. 659–670, 2002.

[16] L. Qian, Y. J. Zhang, and J. W. Huang, "MAPEL: achieving global optimality for a non-convex power control problem," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1553–1563, 2009.

[17] L. Qian, Y. J. Zhang, and M. Chiang, "Globally optimal distributed power control for nonconcave utility maximization," in *Proc. IEEE GLOBECOM*, December 2010, pp. 1–6.

[18] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, 1984.

[19] Y. Chen and M. Chen, "Extended duality for nonlinear programming," *Comput. Optim. Appl.*, vol. 47, no. 1, pp. 33–59, 2010.

[20] S. Kirkpatrick, C. D. Gelatt, and J. M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[22] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[23] A. Eryilmaz, A. Ozdaglar, M. Medard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *IEEE Trans. Inform. Theory*, vol. 54, no. 12, pp. 5511–5524, 2008.

[24] J.-K. Sundararajan, D. Shah, and M. Medard, "ARQ for network coding," in *Proc. IEEE ISIT*, July 2008, pp. 1651–1655.

[25] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 914–925, 2009.

[26] Y. E. Sagduyu and A. Ephremides, "On broadcast stability of queue-based dynamic network coding over erasure channels," *IEEE Trans. Inform. Theory*, vol. 55, no. 12, pp. 5463–5478, 2009.

[27] E. Drinea, C. Fragouli, and L. Keller, "Delay with network coding and feedback," in *Proc. IEEE ISIT*, July 2009, pp. 844–848.

[28] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. IEEE INFOCOM*, April 2009, pp. 208–216.

[29] D. Traskov, M. Medard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. IEEE Globecom Workshops*, Dec. 2009, pp. 1–6.

[30] J. Heide, M. V. Pedersen, F. H. P. Fitzek, and T. Larsen, "Network coding for mobile devices - systematic binary random rateless codes," in *Proc. IEEE ICC Workshops*, June 2009, pp. 1–6.

[31] W. Yeow, A. Hoang, and C. Tham, "Minimizing delay for multicast-streaming in wireless networks with network coding," in *Proc. IEEE INFOCOM*, April 2009, pp. 190–198.

[32] A. A. Yazdi, S. Sorour, S. Valaee, and R. Y. Kim, "Optimum network coding for delay sensitive applications in WiMAX unicast," in *Proc. IEEE INFOCOM*, April 2009, pp. 2576–2580.

[33] A. Fu, P. Sadeghi, and M. Medard, "Delivery delay analysis of network coded wireless broadcast schemes," in *Proc. IEEE WCNC*, April 2012, pp. 2236–2241.

[34] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, 2011.

[35] D. Nguyen and T. Nguyen, "Network coding-based wireless media transmission using POMDP," in *Packet Video Workshop*, May 2009, pp. 1–9.

[36] H. Gangammanavar and A. Eryilmaz, "Dynamic coding and rate-control for serving deadline-constrained traffic over fading channels," in *Proc. IEEE ISIT*, June 2010, pp. 1788–1792.

[37] R. Li and A. Eryilmaz, "Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multi-hop networks," in *Proc. IEEE INFOCOM*, April 2011, pp. 3065–3073.

[38] I.-H. Hou and P. R. Kumar, "Scheduling heterogeneous real-time traffic over fading wireless channels," in *Proc. IEEE INFOCOM*, March 2010, pp. 1–9.

[39] D. P. Bertsekas, *Dynamic Programming and Optimal Control.* Athena Scentific, Belmount, Massachusetts, 2005.

[40] J. J. Jaramillo, R. Srikant, and L. Ying, "Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 979–987, 2011.

[41] L. Yang, Y. Sagduyu, and J. Li, "Adaptive network coding for scheduling real-time traffic with hard deadlines," in *Proc. ACM Mobihoc*, June 2012, pp. 1–10.

[42] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, U.K.: Cambridge Univ. Press, 2004.

[43] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–149, 2006.

[44] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[45] J. Yackoski, B. Azimi-Sadjadi, A. Namazi, J. H. Li, Y. E. Sagduyu, and R. Levy, "RF-NEST: radio frequency network emulator simulator tool," in *Proc. IEEE MILCOM*, Nov. 2011, pp. 1882–1887.

[46] J. Ahrenholz, C. Danilov, T. Henderson, and J. Kim, "CORE: a real-time network emulator," in *Proc. IEEE MILCOM*, Nov. 2008, pp. 1–7.

[47] J. Kiefer, "Sequential minimax search for a maximum," *Proc. Amer. Math. Soc.*, vol. 4, no. 3, pp. 502–506, 1953.

[48] "FCC september commission meeting presentation," September 2009.

[49] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, 2001.

[50] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. on Networking*, vol. 16, no. 2, pp. 396–409, 2008.

[51] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *IEEE/ACM Trans. on Networking*, vol. 15, no. 6, pp. 1333–1344, 2007.

[52] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. IEEE INFOCOM*, April 2006, pp. 1–13.

[53] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in *Proc. IEEE INFOCOM*, April 2009, pp. 486–494.

[54] I.-H. Hou and P. R. Kumar, "Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels," in *Proc. ACM Mobihoc*, May 2009, pp. 175–184.

[55] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *Wireless Networks*, vol. 8, no. 1, pp. 13–26, 2002.

[56] A. Dua and N. Bambos, "Downlink wireless packet scheduling with deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1410–1425, 2007.

[57] S. H. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, "Optimality of myopic sensing in multi-channel opportunistic access," *IEEE Trans. on Information Theory*, vol. 55, no. 9, pp. 4040–4050, 2009.

[58] S. Murugesan, P. Schniter, and N. B. Shroff, "Multiuser scheduling in a Markov-modeled downlink using randomly delayed ARQ feedback," *IEEE Trans. on Information Theory*, vol. 58, no. 2, pp. 1025–1042, 2012.

[59] C. ping Li and M. J. Neely, "Exploiting channel memory for multiuser wireless scheduling without channel measurement: Capacity regions and algorithms," *Performance Evaluation*, vol. 68, no. 8, pp. 631–657, 2011.

[60] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[61] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. on Communications*, vol. 47, no. 11, pp. 1688–1692, 1999.

[62] C. Safran and C. G. Chute, "Exploration and exploitation of clinical databases," *International Journal of Bio-Medical Computing*, vol. 39, no. 1, pp. 151–156, 1995.

[63] D. Wu, D. C. Aliprantis, and K. Gkritza, "Electric energy and power consumption by light-duty plug-in electric vehicles," *IEEE Trans. Power Syst.*, vol. 26, no. 2, pp. 738–746, 2011.

[64] P. Denholm and W. Short, "An evaluation of utility system impacts and benefits of optimally dispatched plug-in hybrid electric vehicles," National Renewable Energy Laboratory, Golden, CO, Tech. Rep. NREL/TP-620-40293, July 2006.

[65] K. Parks, P. Denholm, and T. Markel, "Costs and emissions associated with plug-in hybrid electric vehicle charging in the Xcel Energy Colorado service territory," National Renewable Energy Laboratory, Golden, CO, Tech. Rep. NREL/TP-640-41410, May 2006.

[66] M. Kintner-Meyer, K. Schneider, and R. Pratt, "Impacts assessment of plug-in hybrid vehicles on electric utilities and regional U.S. power grids. Part 1: Technical analysis," Pacific Northwest National Laboratory: Richland, WA, November 2007.

[67] S. W. Hadley and A. A. Tsvetkova, "Potential impacts of plug-in hybrid electric vehicles on regional power generation," *The Electricity Journal*, vol. 22, no. 10, pp. 56 – 68, 2009.

[68] A. Hajimiragha, C. A. Canzares, M. W. Fowler, and A. Elkamel, "Optimal transition to plug-in hybrid electric vehicles in Ontario, Canada, considering the electricity-grid limitations," *IEEE Trans. Ind. Electron.*, vol. 57, no. 2, pp. 690–701, 2010.

[69] J. Taylor, A. Maitra, M. Alexander, D. Brooks, and M. Duvall, "Evaluations of plug-in electric vehicle distribution system impacts," in *Proc. IEEE Power Energy Soc. Gen. Meet.*, July 2010, pp. 1–6.

[70] M. Mahalik, L. Poch, A. Botterud, and A. Vyas, "Impacts of plug-in hybrid electric vehicles on the electric power system in Illinois," in *Proc. IEEE Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES)*, September 2010, pp. 341–348.

[71] R. Sioshansi, R. Fagiani, and V. Marano, "Cost and emissions impacts of plug-in hybrid vehicles on the Ohio power system," *Energy Policy*, vol. 38, no. 11, pp. 6703–6712, 2010.

[72] S. Letendre and R. A. Watts, "Effects of plug-in hybrid electric vehicles on the Vermont electric transmission system," Transport. Res. Board 88th Annu. Meet., Washington DC, 2009.

[73] S. Meliopoulos, J. Meisel, G. Cokkinides, and T. Overbye, "Power system level impacts of plug-in hybrid vechiles," Power Systems Engineering Research Center (PSERC), Tech. Rep., October 2009.

[74] J. Lopes, F. Soares, and P. Almeida, "Integration of electric vehicles in the electric power system," *Proc. IEEE*, vol. 99, no. 1, pp. 168–183, 2011.

[75] K. Schneider, C. Gerkensmeyer, M. Kintner-Meyer, and R. Fletcher, "Impact assessment of plug-in hybrid vehicles on pacific northwest distribution systems," in *Proc. Power Energy Soc. Gen. Meet.*, July 2008, pp. 1–6.

[76] D. Wu, D. C. Aliprantis, and L. Ying, "Load scheduling and dispatch for aggregators of plug-in electric vehicles," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 368–376, 2012.

[77] J. Chow, D. Mello, and K. Cheung, "Electricity market design: an integrated approach to reliability assurance," *Proc. IEEE*, vol. 93, no. 11, pp. 1956–1969, 2005.

[78] R. V. Slyke and R. J. Wets, "L-shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal on Applied Mathematics*, vol. 17, no. 4, pp. 638–663, 1969.

[79] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 371–380, 2010.

[80] E. Sortomme, M. Hindi, S. MacPherson, and S. Venkata, "Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 198–205, 2011.

[81] Q. Li, T. Cui, R. Negi, F. Franchetti, and M. D. Ilic, "On-line decentralized charging of plug-in electric vehicles in power systems," arXiv:1106.5063v2, 2011.

[82] S. Deilami, A. Masoum, P. Moses, and M. Masoum, "Real-time coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile," *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 456–467, 2011.

[83] K. Turitsyn, N. Sinitsyn, S. Backhaus, and M. Chertkov, "Robust broadcast-communication control of electric vehicle charging," in *Proc. IEEE SmartGridComm*, October 2010, pp. 203–207.

[84] M. He, S. Murugesan, and J. Zhang, "Multiple timescale dispatch and scheduling for stochastic reliability in smart grids with wind generation integration," in *Proc. IEEE INFOCOM*, April 2011, pp. 461–465.

[85] J. L. Higle, "Stochastic programming: optimization when uncertainty matters," *Tutorials in Operations Research-INFORMS*, 2005.

[86] U.S. Department of Transportation, Federal Highway Administration, 2009 National Household Travel Survey. [Online]. Available: http://nhts.ornl.gov

[87] A. Brooks, E. Lu, D. Reicher, C. Spirakis, and B. Weihl, "Demand dispatch," *IEEE Power Energy Mag.*, vol. 8, no. 3, pp. 20–29, 2010.

[88] U. E. P. Agency, 2011 electric vehicles, fuel economy. [Online]. Available: http://www.fueleconomy.gov/feg/evsbs.shtml

[89] ISO New England, Day-Ahead Hourly Locational Marginal Prices. [Online]. Available: http://www.iso-ne.com/markets/hst_rpts/hstRpts.do

[90] F. Bouffard and F. Galiana, "Stochastic security for operations planning with significant wind power generation," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 306–316, 2008.

[91] "Accommodating high levels of variable generation," NERC Special Report, www.nerc.com/files/IVGTF_Report_041609.pdf, Apr. 2009.

[92] J. Morales, A. Conejo, and J. Perez-Ruiz, "Economic valuation of reserves in power systems with high penetration of wind power," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 900–910, 2009.

[93] M. Ortega-Vazquez and D. Kirschen, "Estimating the spinning reserve requirements in systems with significant wind power generation penetration," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 114–124, 2009.

[94] S. Murugesan, J. Zhang, and V. Vittal, "Finite state Markov chain model for wind generation forecast: A data-driven spatio-temporal approach," in *Innovative Smart Grid Technologies, IEEE PES*, Jan. 2012, pp. 1–8.

[95] M. He, L. Yang, J. Zhang, and V. Vittal, "A spatio-temporal analysis approach for short-term wind-farm power generation forecast," *submitted to IEEE Trans. Power Syst.*, Sep. 2012.

[96] C. Grigg *et al.*, "The IEEE reliability test system-1996," *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 1010–1020, 1999.

[97] B. J. Kirby, "Spinning reserve from responsive loads," Oak Ridge National Laboratory, Mar. 2003.

[98] J. Black, J. de Bedout, and R. Tyagi, "Incorporating demand resources into optimal dispatch," in *Energy 2030 Conference, 2008. ENERGY 2008. IEEE*, Nov. 2008, pp. 1–8.

[99] A. Papavasiliou and S. S. Oren, "Integration of contracted renewable energy and spot market supply to serve flexible loads," in *18th World Congress of the International Federation of Automatic Control*, Aug. 2011, pp. 1–8.

[100] C. De Jonghe, B. Hobbs, and R. Belmans, "Optimal generation mix with short-term demand response and wind penetration," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 830–839, 2012.

[101] R. Sioshansi and W. Short, "Evaluating the impacts of real-time pricing on the usage of wind generation," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 516–524, 2009.

[102] R. Sioshansi, "Evaluating the impacts of real-time pricing on the cost and value of wind generation," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 741–748, 2010.

[103] G. Strbac, "Demand-side view of electricity markets," *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 520–527, 2003.

[104] P. A. Ruiz, C. R. Philbrick, and P. W. Sauer, "Wind power day-ahead uncertainty management through stochastic unit commitment policies," in *Power Systems Conference and Exposition*, March 2009, pp. 1–9.

[105] A. Papavasiliou, S. S. Oren, and R. P. ONeill, "Reserve requirements for wind power integration: A scenario-based stochastic programming framework," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2197–2206, 2011.

[106] J. F. Restrepo and F. D. Galiana, "Assessing the yearly impact of wind power through a new hybrid deterministic/stochastic unit commitment," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 401–410, 2011.

[107] E. M. Constantinescu, V. M. Zavala, M. Rocklin, S. Lee, and M. Anitescu, "A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 431–441, 2011.

[108] J. Wang, A. Botterud, V. Miranda, C. Monteiro, and G. Sheble, "Impact of wind power forecasting on unit commitment and dispatch," in *Proc. 8th Int. Workshop Large-Scale Integration of Wind Power into Power Systems*, Oct. 2009, pp. 1–8.

[109] "(1998) voluntary load curtailment program. power pool of Alberta." *Available online:http://www.powerpool.ab.ca*.

[110] C. S. Chen and J. T. Leu, "Interruptible load control for Taiwan power company," *IEEE Trans. Power Syst.*, vol. 5, no. 2, pp. 460–465, 1990.

[111] M. Carrion and J. M. Arroyo, "A computationally efficient mixed integer linear formulation for the thermal unit commitment problem," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1371–1378, 2006.

[112] "Western wind and solar integration study," *National Renewable Energy Laboratory, Tech. Rep.*, May 2010.

[113] K. Hedman, M. Ferris, R. O'Neill, E. Fisher, and S. Oren, "Co-optimization of generation unit commitment and transmission switching with N-1 reliability," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 1052–1063, 2010.

[114] Georgia Tech Power Systems Control and Automation Laboratory. Avaialble online:http://pscal.ece.gatech.edu/testsys/index.html.

[115] J. L. Torres, A. Garcia, M. D. Blas, and A. D. Francisco, "Forecast of hourly average wind speed with ARMA models in Navarre (Spain)," *Solar Energy*, vol. 79, no. 1, pp. 65–77, 2005.

[116] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[117] X. Zhou and H. Zheng, "TRUST: a general framework for truthful double spectrum auctions," in *Proc. IEEE INFOCOM*, April 2009, pp. 999–1007.

[118] J. Jia, Q. Zhang, Q. Zhang, and M. Liu, "Revenue generation for truthful spectrum auction in dynamic spectrum access," in *Proc. ACM MobiHoc*, May 2009, pp. 3–12.

[119] K. Ryan, E. Arvantinos, and M. Buddhikot, "A new pricing model for next generation spectrum access," in *Proc. TAPAS*, August 2006.

[120] S. Sengupta and M. Chatterjee, "An economic framework for dynamic spectrum access and service pricing," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1200–1213, 2009.

[121] S. Gandhi, C. Buragohain, L. Cao, H. Zheng, and S. Suri, "Towards real-time dynamic spectrum auctions," *Computer Networks (Elsevier) Special Issue on Cognitive Wireless Networks*, vol. 52, no. 5, pp. 879–897, 2008.

[122] Y. Wu, B. Wang, K. J. R. Liu, and T. C. Clancy, "A multi-winner cognitive spectrum auction framework with collusion-resistant mechanisms," in *Proc. IEEE DySPAN*, October 2008, pp. 1–9.

[123] J. Huang, R. A. Berry, and M. L. Honig, "Auction-based spectrum sharing," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 405–418, 2006.

[124] H. Mutlu, M. Alanyali, and D. Starobinski, "Spot pricing of secondary spectrum usage in wireless cellular networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1794–1804, 2008.

[125] D. Niyato, E. Hossain, and Z. Han, "Dynamic spectrum access in IEEE 802.22-based cognitive wireless networks: a game theoretic model for

competitive spectrum bidding and pricing," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 16–23, 2009.

[126] Y. Xing, R. Chandramouli, and C. M. Cordeiro, "Price dynamics in competitive agile spectrum access markets," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 613–621, 2007.

[127] O. Ileri, D. Samardzija, T. Sizer, and N. B. Mandayam, "Demand responsive pricing and competitive spectrum allocation via a spectrum server," in *Proc. IEEE DySPAN*, November 2005, pp. 194–202.

[128] D. Niyato, E. Hossain, and Z. Han, "Dynamics of multiple-seller and multiple-buyer spectrum trading in cognitive radio networks: a game-theoretic modeling approach," *IEEE Trans. Mobile Comput.*, vol. 8, no. 8, pp. 1009–1022, 2009.

[129] D. Xu, X. Liu, and Z. Han, "A two-tier market for decentralized dynamic spectrum access in cognitive radio networks," in *Proc. IEEE SECON*, June 2010, pp. 1–9.

[130] L. Duan, J. Huang, and B. Shou, "Investment and pricing with spectrum uncertainty: a cognitive operator's perspective," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1590–1604, 2011.

[131] S. Wang, J. Zhang, and L. Tong, "Delay analysis for cognitive radio networks with random access: a fluid queue view," in *Proc. IEEE INFOCOM*, March 2010, pp. 1–9.

[132] ——, "A characterization of delay performance of cognitive medium access," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 800–809, 2012.

[133] Q. Chen, Y.-C. Liang, M. Motani, and W. C. Wong, "CR-CSMA: a random access MAC protocol for cognitive radio networks," in *Proc. IEEE PIMRC*, September 2009, pp. 486–490.

[134] D. P. Bertsekas and R. Gallager, *Date Networks.* Prentice Hall, Englewood Cliffs, NJ, 1990.

[135] P. Hande, M. Chiang, A. R. Calderbank, and J. Zhang, "Pricing under constraints in access networks: revenue maximization and congestion management," in *Proc. IEEE INFOCOM*, March 2010, pp. 1–9.

[136] A. Mas-Colell, M. Whinston, and J. Green, *Microeconomic Theory*. Oxford University Press, Oxford, United Kingdom, 1995.

[137] M. J. Osborne and A. Rubinstein, *A Course In Game Theory*. The MIT Press, 1994.

[138] J. Sun and E. Modiano, "Channel allocation using pricing in satellite networks," in *Proc. CISS*, March 2006, pp. 182–187.

[139] D. Zheng, W. Ge, and J. Zhang, "Distributed opportunistic scheduling for ad-hoc networks with random access: an optimal stopping approach," *IEEE Trans. Inf. Theory*, vol. 55, no. 1, pp. 205–222, 2009.

[140] J. Papandriopoulos, S. Dey, and J. Evans, "Optimal and distributed protocols for cross-layer design of physical and transport layers in MANETs," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1392–1405, 2008.

[141] J. Papandriopoulos and J. S. Evans, "SCALE: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks," *IEEE/ACM Trans. Netw.*, vol. 55, no. 8, pp. 3711–3724, 2009.

[142] Y. Xu, T. Le-Ngoc, and S. Panigrahi, "Global concave minimization for optimal spectrum balancing in multi-user DSL networks," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2875–2885, 2008.

[143] R. Horst and H. Tuy, *Global Optimization - Deterministic Approaches*. Springer, New York, 2003.

[144] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, no. 2, pp. 311–329, 1988.

[145] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[146] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, 2005.

[147] H.-W. Lee, E. Modiano, and L. B. Le, "Distributed throughput maximization in wireless networks via random power allocation," *IEEE Trans. Mobile Comput.*, vol. 11, no. 4, pp. 577–590, 2011.

[148] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, March 2000, pp. 585–594.

[149] K. Wang, C. F. Chiasserini, J. G. Proakis, and R. R. Rao, "Joint scheduling and power control supporting multicasting in wireless ad hoc networks," *Ad Hoc Networks*, vol. 4, no. 4, pp. 532–546, 2006.

[150] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 6, no. 5, pp. 556–567, 2000.

[151] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, 1995.

[152] X. Lin, N. B. Shroff, and R. Srikant, "On the connection-level stability of congestion-controlled communication networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2317–2338, 2008.

[153] R. Rao and A. Ephremides, "On the stability of interacting queues in a multiple-access system," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, pp. 918–930, 1988.

[154] A. ParandehGheibi, M. Medard, A. Ozdaglar, and A. Eryilmaz, "Information theory vs. queueing theory for resource allocation in multiple access channels," in *Proc. IEEE Pers. Ind. Mob. Radio Commun.*, September 2008, pp. 1–5.

[155] R. Loynes, "The stability of a queue with non-interdependent interarrival and service times," *Proc. Camb. Philos. Soc.*, vol. 58, pp. 497–520, 1962.

[156] P. Chaporkar and S. Sarkar, "Stable scheduling policies for maximizing throughput in generalized constrained queueing systems," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1913–1931, 2008.

[157] Y. Yi, J. Zhang, and M. Chiang, "Delay and effective throughput of wireless scheduling in heavy traffic regimes: vacation model for complexity," in *Proc. ACM Mobihoc*, May 2009, pp. 55–64.